The Dissertation Committee for Sreangsu Acharyya

certifies that this is the approved version of the following dissertation:

Learning to Rank in Supervised and Unsupervised Settings using Convexity and Monotonicity

Committee:

Joydeep Ghosh, Supervisor

Sanjay Shakkottai

Constantine Caramanis

Sriram Vishwanath

Inderjit Dhillon

Learning to Rank in Supervised and Unsupervised Settings using Convexity and Monotonicity

by

Sreangsu Acharyya, B.E.; M.Tech

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2013

To my dear brother Swarnangsu (Tatu) and parents Subhrangsu Acharyya and Sumitra Acharyya.

Acknowledgments

It takes a village to raise a child, goes the proverb. Surprisingly, this dissertation has needed the same.

First I would like to thank Prof. Joydeep Ghosh, for his support, for persisting with me over the long period of my PhD and for making a case for me many a time, without which my extended stay would not have been possible. I am an improved writer thanks to his painstaking corrections and guidance. Prof. Inderjit Dhillon with his lucid yet detailed teaching style, that is infectiously engaging, got me interested in ranking in the first place. I have gained much clarity from Prof. Constantine Caramanis's questions about my work and his stimulating course on convex analysis. I would like to thank Prof. Sanjay Shakkottai and Prof. Sriram Vishwanath for serving in my committee and for their encouragement, and advice.

I will remember my time in the IDEAL lab with a lot of fondness. I thank Oluwasanmi Koyejo for being my partner in crime in writing papers and code together. Suriya Gunasekar for always being cheerfully eager to help and amuse, for reading and commenting on my half baked drafts and dragging/rolling us (literally, sometimes when seated on our lab chairs !) outside of the lab every once in a while. Sytem administering the lab machines has been a lot of fun with Chase Krumpleman, Clinton Jones, Cheng Lee and Yubin Park. As has been to enlist Ayan Acharyya, Joyce Hoi, Rajiv Khanna in our regular pranks (Suriya being mostly ate the receiving end). The lab would not be the same with our vigorous and entertaining discussion with Virag Shah and Vinay Joseph. I thank Virag, R. Suju, Sandip

Ray (Mama) and Shameek Sinha for hosting me at times that I did not have an apartment. I do not think it is possible for anyone to be more helpful than Mama. I do miss the one word telephone calls at 3:30 AM ”Metro ?”, that had become the signature of our midnight-early-morning coffee drinking ritual.

I think one learns the most from one’s peers. I have benefitted immensely from discussions with Priyank Patel, Arindam Banerjee, Mama and Srujana Merugu. They have all been an oracle as well as a patient ear.

I have gained not one but two families and homes away from home. One in, AID-AUSTIN and another our own institute of excessively eating engineers. Without our regular meetings all over U.S. (missing flights notwithstanding) with Arindam, Anirban, Sujata, Sandhitsu and Siddharta participating from remote. Things would have not been so enjoyable. Same with Madhulika, Chandrika and Kavita for making me a part of their AID-Austin family, our regular saturday evening hangouts. Saikat Maitra and Sayan Saha have been great room mates.

I would like to thank Sathiya Keerthi, Phil Bohannon, Srujana Merugu, Kunal Punera and R.Suju for my enriching time at Yahoo!. In particular I have learned a lot for my interaction with Sathiya Keerthi. I thank Vijay Boyapatiand Tom Annau for stimulating and extremely entertaining internships at Google. L.V.Subramaniam, Sumit Negi and Shourya Roy for the great time at IRL,India. Nena Marin and Srivatsava for our brief but intense stint of midnight hacking on KDD-Cup.

However, nothing would have been possible without the unflinching support of my parents, Dr. Subhranhgsu Acharyya and Sumitra Acharyya and without them being inspirational examples to emulate, showing me to do the right thing but always by example. Same with my brother Swarnangsu to the extent that it puzzles me who is the elder among us. Many stepped in to fill the gap that I had left because of my extended absence from home. Stepped in when I was sorely needed in very distressing times. I cannot express that gratitude and I dare not name them, my didis, aunt and uncles. Payal, if you are reading this,

you have been a botomless well of indulgent patience, without which all this would have been possible.

SREANGSU ACHARYYA

The University of Texas at Austin
August 2013

Learning to Rank in Supervised and Unsupervised Settings using Convexity and Monotonicity

Publication No. _____

Sreangsu Acharyya, Ph.D.
The University of Texas at Austin, 2013

Supervisor: Joydeep Ghosh

This dissertation addresses the task of learning to rank, both in the supervised and unsupervised settings, by exploiting the interplay of convex functions, monotonic mappings and their fixed points. In the supervised setting of learning to rank, one wishes to learn from examples of correctly ordered items whereas in the unsupervised setting, one tries to maximize some quantitatively defined characteristic of a "good" ranking.

A ranking method selects one permutation from among the combinatorially many permutations defined on the items to rank. Accomplishing this optimally in the supervised setting, with minimal loss in generality, if any, is challenging. In this dissertation this problem is addressed by optimizing, globally and efficiently, a statistically consistent loss functional over the class of compositions of a linear function by an arbitrary, strictly

monotonic, separable mapping with large margins. This capability also enables learning the parameters of a generalized linear model with an unknown link function. The method can handle infinite dimensional feature spaces if the corresponding kernel function is known.

In the unsupervised setting, a popular ranking approach is is link analysis over a graph of recommendations, as exemplified by pagerank. This dissertation shows that pagerank may be viewed as an instance of an unsupervised consensus optimization problem. The dissertation then solves a more general problem of unsupervised consensus over noisy, directed recommendation graphs that have uncertainty over the set of "out" edges that emanate from a vertex. The proposed consensus rank is essentially the pagerank over the *expected* edge-set, where the expectation is computed over the distribution that achieves the most agreeable consensus. This consensus is measured geometrically by a suitable Bregman divergence between the consensus rank and the ranks induced by item specific distributions

Real world deployed ranking methods need to be resistant to spam, a particularly sophisticated type of which is link-spam. A popular class of countermeasures "de-spam" the corrupted webgraph by removing abusive pages identified by supervised learning. Since exhaustive detection and neutralization is infeasible, there is a need for ranking functions that can, on one hand, attenuate the effects of link-spam without supervision and on the other hand, counter spam more aggressively when supervision is available. A family of non-linear, iteratively defined monotonic functions is proposed that propagates "rank" and "trust" scores through the webgraph. It relies on non-linearity, monotonicity and Schur-convexity to provide the resistance against spam.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many applications, such as information retrieval and recommender systems, require items to be ordered according to user preference. Usually, the "score" that defines the *transitive relation* of order among the items is unavailable and only the sorted order of training items can be observed. This inaccessibility motivates the learning to rank (LETOR) problem. In the supervised setting the learner has access to representative examples of correctly ordered items from which it is expected to minimize the number of ordering "mistakes".

In general, a LETOR problem consists of a set of queries $\mathcal{Q} = \{q_1, q_i \dots q_{|\mathcal{Q}|}\}$ and a set of items $\mathcal{V}$ that are to be ranked in the context of the queries. For every query q_i, there is a subset $\mathcal{V}_i \subset \mathcal{V}$ whose elements have been ordered, based on their relevance. This ordering is customarily expressed via a rank score vector $\tilde{\boldsymbol{r}}_i \in \mathbb{R}^{d_i=|\mathcal{V}_i|}$ whose components $\tilde{r}_{ij}$ correspond to the score of the j^{th} items. In some cases the actual values of $\tilde{r}_{ij}$ are of no significance except for establishing an order over the set $\mathcal{V}_i$. In this case the problem becomes that of predicting a permutation. In this dissertation we distinguish the learning to rank task from a related one of learning binary pairwise relations where transitivity is not required.What differentiates learning to rank (LETOR) from other prediction problems, e.g. classification and regression is this combinatorial structure of the output space.

Existing LETOR techniques fall in the following 3 categories:

1. point-wise,

2. pair-wise

3. list-wise methods.

In point-wise methods, the higher ranked items are assigned higher target scores. These methods then ignore the structure and solve a regression problem. Pair-wise methods capture some structure by posing the task as a classification problem over all pairs. However, this results in a quadratic growth in the training set, often ameliorated by down-sampling. However, pairwise-methods also suffer from insufficient structure: their predictions need not obey transitivity. An *order-reconciliation step* is necessary for predicting ordered outputs which is NP hard Cohen et al. (1999), necessitating approximations and heuristics. List-wise methods wrestle with the full combinatorial structure and thus have to deal with formidable optimization problems. Typically, they have to cut corners using sampling (Weston and Blitzer, 2012) and or approximations (Ailon and Mohri, 2008) to make the algorithms scale.

Many cost functions have been designed to evaluate rankings, e.g. (normalized) discounted cumulative gain ((N)DCG), (Järvelin and Kekäläinen, 2000), expected reciprocal rank (ERR) (Chapelle et al., 2009), mean average precision (MAP) (Baeza-Yates and Ribeiro-Neto, 1999), etc. Implicitly or explicitly, these are functions over permutations. They are reasonably easy to compute given a ranking, but hard to train on because they lead to difficult combinatorial problems.

An ideal LETOR formulation should (i) capture the combinatorial structure like the list-wise methods, but with (ii) algorithms that are no more complicated than point-wise methods. While this seems too much to ask for, this dissertation makes some progress in that direction. The dissertation uses a flexible family of statistically consistent, efficiently optimize-able cost functions capturing the desirable characteristics of ranking.

Both supervised and unsupervised techniques are addressed in this dissertation. Supervised learning algorithms for ranking require representative examples of correctly or-

dered items. Obtaining this information can be quite expensive. So it is important to have complementary techniques that do not need training examples. In the unsupervised setting, algorithms do not receive information about how the set of training items should be ranked. Typically they exploit some axiomatic characterization of order among items, for example, an unsupervised paradigm that has been very successful in ranking items based on a graph of recommendations is link analysis. Pagerank (Brin and Page, 1998) and HITS (Kleinberg, 1999a) are two of the most well known algorithms in this category. They view the graph $\mathcal{G}$ as a distributed recommendation system where each vertex recommends other vertices through its out-edges (directed edges that leave the vertex). However these algorithms are (i) susceptible to spam and (ii) do not incorporate fluctuations in the edge set of the graph. This dissertation explores convexity and monotonicity based approaches to incorporate these properties.

Main Contributions

A novel approach for learning to rank (LETOR) based on the notion of monotone retargeting is introduced in **Chapter 3**. Monotone retargeting (MR) minimizes a divergence between all monotonic increasing transformations of the relevance scores and a parameterized prediction function. The novelty lies in the fact that the minimization is over the transformations as well as over the parameters. MR is applied with Bregman divergences, a large class of "distance like" functions that were recently shown to be the unique class that is statistically consistent with the normalized discounted gain (NDCG) criterion (Ravikumar et al., 2011). The algorithm uses alternating projection style updates, in which one set of simultaneous projections can be computed independent of the Bregman divergence and the other projection reduces to parameter estimation of a generalized linear model. This results in an easily implementable and efficiently parallelizable algorithm for the LETOR task that enjoys global optimum guarantees under mild conditions. We present empirical results on benchmark datasets showing that this approach can substantially outperform the

state of the art NDCG consistent techniques.

Tools of convexity and large margins are brought to bear upon the task of learning permutations from examples. This leads to novel and efficient algorithms with guaranteed prediction performance in the online setting and on global optimality and the rate of convergence in the batch setting. As a result, an effective algorithm is obtained to learn transitive relationship over items. It captures the inherent combinatorial characteristic of the output space yet it has a computational burden not much more than a generalized linear model.

Statistical consistency of different LETOR algorithms with respect to ranking quality metrics is an active area of research. Ravikumar et al. (2011) identify and exhaustively characterize the cost functions that are consistent with respect to NDCG, a popular rank quality metric. This turns out to be the loglikelihood of canonical generalized linear models (McCulloch and Searle, 2001), a traditional technique of parametric regression popular among statisticians and machine learners alike. Each member of this family is characterized by a finite dimensional vector that needs to be estimated from data. A natural question to ask is whether it is possible to search not only over the parameters but also over all members of the family. Note that this entails a search over all monotonic functions, or equivalently all convex functions. **Chapter 4** of this dissertation introduces efficient techniques for this purpose. The difference of this model from that pursued in Chapter 3 is that the loss function and the monotonic transform are tied to each other, this coupling leads to guarantees of joint convexity. The added generality of simultaneously optimizing over monotonic functions and parameters comes only at an extra cost of $\log d$ where d is the dimensionality of the data.

An unsupervised method is proposed in **Chapter 5** to solve a consensus ranking problem defined over noisy, directed recommendation graphs. In these noisy directed graphs, the edge weights indicate endorsement of a vertex by another but there is uncertainty over the set of "out" edges that emanate from a vertex. This uncertainty is modeled by weights over the discrete set of such possible "out" edge-sets associated with every ver-

tex. Pagerank induces a ranking over the vertices of a graph for a particular choice of an "out" edge-set, whereas the proposed method combines the multiple rankings that could be induced by the different choices. The proposed consensus rank is essentially the pagerank over the *expected* edge-set, where the expectation is computed over the distribution that achieves the most agreeable consensus. The consensus is measured geometrically by a suitable Bregman divergence between the consensus rank and the ranks induced by the pure distributions [1] over the choices of the "out" edge-sets. The practice of ranking vertices by the stationary distribution of a random walk over a *noise-free* graph is extended to *noisy* graphs. The method can be applied to (multi-)graphs with (i) different types of labeled edges whose label weights are unknown, (ii) per vertex edge sets known to lie in a polyhedron of uncertainty, possibly defined by partial order constraints. Two families of algorithms are provided to solve this optimization problem by exploiting new results concerning Bregman divergences that were derived for this purpose.

Finally, **Chapter 6** deals with spam resistance. The ranking scheme of a search engine needs to be resistant to spam, a particularly sophisticated type of which is link-spam. Current countermeasures "de-spam" the corrupted webgraph by removing abusive pages identified by supervised learning. Since exhaustive detection and neutralization is infeasible, there is a need for ranking functions that can, on one hand, attenuate the effects of link-spam without supervision and on the other hand, counter spam more aggressively when supervision is available. A family of non-linear functions is proposed that propagate "rank" and "trust" scores through the webgraph. It includes Pagerank as a special case and relies on non-linearity, monotonicity and Schur-convexity to provide spam resistance. The main contributions here are (i) the proof of convergence and uniqueness of the iterates, and (ii) empirical comparison with Pagerank and other established anti-spam rankings.

[1] distributions over a discrete set concentrated fully on one item.

Chapter 2

Background

In this chapter we give a brief summary of convexity and properties of Bregman divergences that recur throughout the dissertation.

Notation: Vectors are denoted by bold lower case letters. The i_{th} component of the vector $\boldsymbol{x}$ is indicated by x_i. When suitable, we also indicate the *entire* vector $\boldsymbol{x}$ by decorating its i^{th} component as follows: $\vec{x_i}$. This form is used to convey succinctly how a vector has been constructed from its components. The symbol $T^\dagger$ indicates the transpose of matrix T. Random variables are also indicated by capital letters. $\mathop{\mathbb{E}}_{X\sim\boldsymbol{p}}[f(X)]$ represents the expectation of a function $f(\cdot)$ of a random variable X having a distribution $\boldsymbol{p}$. Sets are denoted by (matching) calligraphic letters, for instance random variable X takes values in a set $\mathcal{X}$. The unit simplex is denoted by Δ, its dimensionality will be implicit. For the most part we deal only with sets in the Euclidean vector space $\mathbb{R}^d$. The notation ${\mathbb{R}_+}^d$ will denote the positive orthant of $\mathbb{R}^d$, and $\mathbb{R}^d_\epsilon$ will denote the set $\{\boldsymbol{x}|\boldsymbol{x} \in \mathbb{R}^d \cap x_i > \epsilon \ \forall_i\}$, whereas the symbol Δ_ϵ will indicate the set $\{\boldsymbol{x}|\boldsymbol{x} \in \Delta \cap x_i > \epsilon \ \forall_i\}$ and the symbol $\blacktriangle$, the set $\{\boldsymbol{x}|\sum_i x_i \leq 1 \ \boldsymbol{x} \in \mathbb{R}_+\}$. Familiarity with convex analysis is assumed.

2.1 Convex Analysis Review

This section is a brief review of convex analytic notions that are used in the dissertation. A function is **convex** if the following inequality holds for any points $\boldsymbol{x}, \boldsymbol{y}$ in its domain:

$$\phi(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha\phi(\boldsymbol{x}) + (1-\alpha)\phi(\boldsymbol{y}).$$

The function is **strictly convex** if the previous inequality is strict. It has **modulus of strong convexity** s if the following inequality holds:

$$\phi(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha\phi(\boldsymbol{x}) + (1-\alpha)\phi(\boldsymbol{y}) - \frac{s}{2}\alpha(1-\alpha)||\boldsymbol{x}-\boldsymbol{y}||^2, \tag{2.1}$$

which for differentiable $\phi(\cdot)$ is equivalent to:

$$\langle \nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y}), x - y\rangle \geq s||\boldsymbol{x}-\boldsymbol{y}||^2. \tag{2.2}$$

For a twice differentiable $\phi(\boldsymbol{x})$, this means that eigenvalues of its Hessian are lower bounded by s.

The **epigraph** of the function ϕ is the set $\{(\boldsymbol{x}, y) \mid y \geq \phi(\boldsymbol{x})\}$. The **sub-level** set of the function ϕ for the level γ is a set $\{\boldsymbol{x} \mid \phi(\boldsymbol{x}) \leq \gamma\}$. The function is defined to be **closed** (equivalently lower semi–continuous) if its epigraph is closed, as a consequence the sub level sets are closed as well. A convex function ϕ is **proper** if $\operatorname{dom}\phi$ is non-empty and $\forall \boldsymbol{x} \in \operatorname{dom}\phi$ s.t. $\phi(x) > -\infty$.

The **Legendre conjugate** $\psi(\cdot)$ of the function $\phi(\cdot)$ is defined as

$$(\phi)^*(\boldsymbol{\lambda}) \triangleq \psi(\boldsymbol{\lambda}) \triangleq \sup_{\boldsymbol{x}}(\langle\boldsymbol{\lambda}, \boldsymbol{x}\rangle - \phi(\boldsymbol{x})).$$

The superscript $*$ when applied to functions will indicate the conjugation operation. If ϕ is closed, proper, strictly convex function, as will always be the case in this paper,

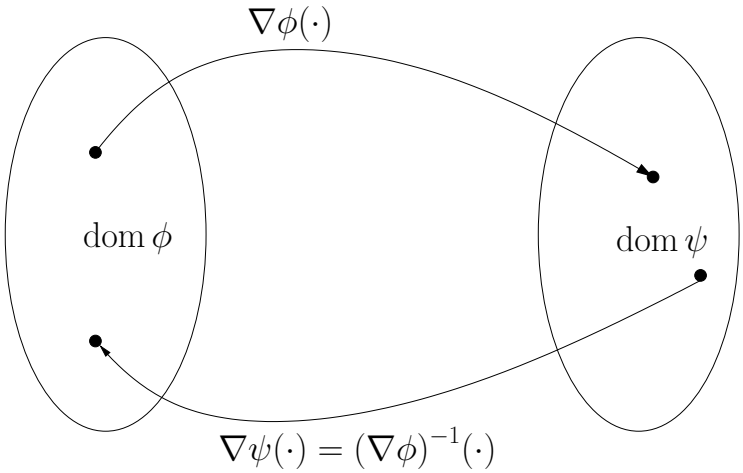


Figure 2.1: The gradient mapping between domains of Legendre conjugate functions

$((\phi(\cdot))^*)^* = \phi(\cdot)$ and $(\nabla\phi(\cdot))^{-1} = \nabla\psi(\cdot)$ is a one to one map (See figure 2.1).

A closed, proper convex function ϕ is of the **Legendre type** if its domain has a non-empty interior and the following holds

- ϕ is strictly convex and differentiable on $\mathrm{int}\,\mathrm{dom}\,\phi$,
- $\forall \boldsymbol{y} \in \mathrm{bd}(\mathrm{dom}\,\phi), \forall \boldsymbol{x} \in \mathrm{int}(\mathrm{dom}\,\phi)$. the limit $\lim_{\boldsymbol{x}\to\boldsymbol{y}} \|\nabla\phi(\boldsymbol{x})\| \to \infty$

In convex analysis, the **indicator function** is defined as as:

$$\delta(x|\mathcal{X}) = \begin{cases} 0 & \text{if } x \in \mathcal{X} \\ \infty & \text{otherwise} \end{cases}$$

It is closed and convex if the set $\mathcal{X}$ is closed and convex. The Legendre dual of the **indicator function** of a closed convex set $\mathcal{X}$ is a sublinear function called the **support function** of the set $\mathcal{X}$. The support function of any set $\mathcal{X}$ is independently defined as

$$\delta^*{}_{\mathcal{X}}(\boldsymbol{s}) \triangleq \sup_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{s} \rangle .$$

If $\mathcal{X}$ is closed and convex then it follows that support function can be used to give a complete

characterization of the set using the property $\mathcal{X} = \{\boldsymbol{x} \mid \langle \boldsymbol{x}, \boldsymbol{s} \rangle \leq \delta^*{}_X(\boldsymbol{s})\}$. All sublinear functions are support functions, as a result there is a one to one correspondence with closed convex sets and sublinear functions.

A non-negative, positively homogeneous, proper function with degree 1 may be obtained from a convex set $\mathcal{Y}$ containing the origin. Such a function is called a **Gauge** and is defined as:

$$\text{Gauge}_{\mathcal{Y}}(\boldsymbol{y}) = \inf\{\lambda \mid \boldsymbol{y} \in \lambda\mathcal{Y}\}.$$

Given a convex function $\phi(x)$ one can define for all $\lambda > 0$ its **perspective function**

$$\pi(\lambda, x) = \lambda\phi(\frac{x}{\lambda}).$$

The function $\pi(\lambda, x)$ when treated as a function of x alone is called the dilation of $\phi(\cdot)$. Both the **dilation** and the **perspective** functions are convex functions. Note however, that some domain qualification may apply that limits the range of values that λ can take.

The **Fenchel-Young** inequality (2.3) is fundamental to convex analysis and plays an important role in our analysis.

$$\psi(\boldsymbol{y}) + \phi(\boldsymbol{x}) - \langle \boldsymbol{y}, \boldsymbol{x} \rangle \geq 0. \tag{2.3}$$

2.2 Bregman Divergence

Definition 1. Bregman Divergence: *Let* $\phi : \Theta \mapsto \mathbb{R},\ \Theta = \operatorname{dom}\phi \subseteq \mathbb{R}^d$ *be a strictly convex, closed function, differentiable on* $\operatorname{int}\Theta$. *For* $\boldsymbol{x} \in \operatorname{dom}(\phi),\ \boldsymbol{y} \in \operatorname{int}\Theta$, *the Bregman divergence* $D_\phi\big(\cdot\big|\big|\cdot\big) : \operatorname{dom}(\phi) \times \operatorname{int}(\operatorname{dom}(\phi)) \mapsto \mathbb{R}_+$ *corresponding to* ϕ, *is defined as*

$$D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \triangleq \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\phi(\boldsymbol{y}) \rangle .$$

It is easy to show that $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \geq 0$ and $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$. As the readers

will notice, Bregman divergences are asymmetric in general and guaranteed to be strictly convex only in the first argument. A convenient identity that helps in analyzing convexity properties with respect to the second argument is:

$$D_\psi\big(\nabla\phi(\boldsymbol{y})\Big|\Big|\nabla\phi(\boldsymbol{x})\big) = D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big). \tag{2.4}$$

We will require a few additional properties of the function ϕ. These are:

P1: $\lim_{\theta\to\theta_b\in\mathrm{bd}(\Theta)} \|\nabla\phi(\theta)\| = \infty$

P2: If sequence $\boldsymbol{x}_t \in \mathrm{int}(\mathrm{dom}\,\phi)$ and $\lim\limits_{t\to\infty} \boldsymbol{x}_t = \boldsymbol{x}$ then $\lim\limits_{t\to\infty} D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{x}_t\big) = 0$

P3: The left sublevel set

- $L_r(\boldsymbol{y}) \triangleq \{\boldsymbol{x}|D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big) < r\}$ is bounded for all $\boldsymbol{y} \in \mathrm{int}\,\Theta$

In this dissertation, we only consider functions of the form $\boldsymbol{\phi}(\cdot) : \mathbb{R}^n \ni \boldsymbol{x} \mapsto \sum_i w_i\phi(x_i)$ which are weighted sums of *identical* scalar convex functions applied to each component. We refer to this class as *weighted, identically separable* (**WIS**) or simply **IS** if the weights are equal. This class has properties particularly suited to ranking. Mahalonobis distance with diagonal W, weighted KL divergence $wKL\left(\boldsymbol{x}\|\boldsymbol{y}\right)$ and weighted and shifted generalized I-divergence $wGI\left(\boldsymbol{x}\|\boldsymbol{y}\right)$ are in this family (Table 3.1).

When the interior of the domain of the function ϕ is empty special care is required to define the Bregman divergence because the gradient as it is usually defined does not exist. In an ϵ neighborhood of a point in the relative interior of the function, the value of the function is finite on the intersection of this neighborhood with the affine hull of the domain but infinite at other points of the neighborhood, thus making the function non-differentiable in the customary sense. It is however possible to define a linear function on the affine hull of the domain that approximates the convex function in its *relative neighborhood*, leading to the notion of *relative gradient*.

Consider the restriction ϕ_r of an everywhere defined function ϕ that is convex on an affine subset $\mathcal{A}$ of its domain, defined as follows:

$$\phi_r(\boldsymbol{x}) = \begin{cases} \phi(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \mathcal{A} \subset \operatorname{dom}\phi \\ \infty & \text{otherwise} \end{cases}.$$

The symbol $\mathcal{A}_{\|}$ denotes the subspace parallel to the affine set $\mathcal{A}$. Using the property

$$\langle \nabla\phi(\boldsymbol{x}), \boldsymbol{d}\rangle = \left\langle \operatorname{Proj}_{\mathcal{A}_{\|}}(\nabla\phi(\boldsymbol{x})), \boldsymbol{d}\right\rangle \ \forall \boldsymbol{d} \in \mathcal{A}_{\|}$$

one may define the relative gradient of the function ϕ_r as

$$\nabla_{\mathrm{ri}}\phi_r(\boldsymbol{x}) \triangleq \operatorname{Proj}_{\mathcal{A}_{\|}}(\nabla\phi(\boldsymbol{x}))$$

and a relative inner product

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{A}_{\|}} = \left\langle \operatorname{Proj}_{\mathcal{A}_{\|}}(\boldsymbol{x}), \operatorname{Proj}_{\mathcal{A}_{\|}}(\boldsymbol{y})\right\rangle. \tag{2.5}$$

Definition 2. Bregman Divergence(with Empty Interior): *Let* $\phi : \Theta \mapsto \mathbb{R},\ \Theta = \operatorname{dom}\phi \subseteq \mathbb{R}^d$ *be a strictly convex, closed function, relatively differentiable on* $\operatorname{ri}\operatorname{int}\Theta$. *For* $\boldsymbol{x} \in \operatorname{dom}(\phi),\ \boldsymbol{y} \in \operatorname{ri}\operatorname{int}\Theta$, *the Bregman divergence* $D_\phi\big(\cdot\big|\big|\cdot\big) : \operatorname{dom}(\phi)\times\operatorname{ri}\operatorname{int}(\operatorname{dom}(\phi)) \mapsto \mathbb{R}_+$ *corresponding to* ϕ, *is defined as* $D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big) \triangleq \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x}-\boldsymbol{y}, \nabla_{\mathrm{ri}}\phi(\boldsymbol{y})\rangle_{\operatorname{ri}\operatorname{int}\operatorname{dom}\phi_{\|}}$.

Example 1. *Consider the Bregman divergence obtained by the function*

$$\phi(\boldsymbol{p}) = \begin{cases} \sum_i (p_i \log p_i - p_i) & \textit{for } \boldsymbol{p} \in \Delta \subset \mathbb{R}^n \\ +\infty & \textit{otherwise}. \end{cases} \tag{2.6}$$

The function is closed, strictly convex and differentiable in its relative interior, *with the gradient of* $\sum_i (p_i \log p_i - p_i)$ *given by* $\vec{\log} p_i$. *The relative gradient* $\nabla_{\mathrm{ri}}\phi$ *can be obtained*

by projecting the gradient $\vec{\log} p_i$ *on the subspace parallel to the affine hull of* Δ *which is the set* $\mathcal{A} = \{\boldsymbol{x} | \langle \mathbf{1}, \boldsymbol{x} \rangle = 0\}$. *Thus*

$$\nabla_{\mathrm{ri}}\phi(\boldsymbol{p}) \triangleq \mathrm{Argmin}_{\boldsymbol{v}\in\mathcal{A}} \|\vec{v_i} - \vec{\log} p_i\|_2^2 = \begin{bmatrix} \vdots \\ \log p_i - \lambda \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \log p_i - \frac{1}{n}\sum_i^n \log p_i \\ \vdots \end{bmatrix}.$$

Here λ *is the Lagrange multiplier enforcing the constraint. Note that the*

$$\lim_{\boldsymbol{p}\to \mathrm{ri}(\mathrm{bd}(\mathrm{dom}\,\phi))} \|\nabla_{\mathrm{ri}}\phi(\boldsymbol{p})\| = \infty.$$

Also, given a vector $\boldsymbol{y}$ *as the relative gradient one may invert* ∇_{ri} *to obtain*

$$\begin{aligned} \boldsymbol{p} = (\nabla_{\mathrm{ri}}\phi(\boldsymbol{y}))^{-1} &= \begin{bmatrix} \vdots \\ \frac{e^{y_i}}{\sum_i e^{y_i}} \\ \vdots \end{bmatrix} = \nabla \log \left(\sum_i e^{y_i} \right) \\ &= \nabla_{\boldsymbol{y}} \left[\max_{\boldsymbol{p}\in\Delta} \langle \boldsymbol{y}, \boldsymbol{p} \rangle - \phi(\boldsymbol{p}) \right] = \nabla_{\boldsymbol{y}} \phi^*(\boldsymbol{y}). \end{aligned} \tag{2.7}$$

[1] *Using definition* (2) *we obtain the corresponding Bregman divergence between* $\boldsymbol{p}, \boldsymbol{q} \in \Delta$

[1] Particularly important is that the image of the simplex Δ with respect to the relative gradient is whole of $\mathbb{R}^n$ and convex, whereas the image with respect to the gradient $\vec{\log} p_i$ is not. The image of Δ with respect to $\nabla_{\mathrm{ri}}\phi(\boldsymbol{p})$ is also the domain of the Legendre dual ϕ^*.

as

$$
\begin{aligned}
D_\phi\big(\boldsymbol{p}\big|\big|\boldsymbol{q}\big) &= \sum_i^n (p_i \log p_i - p_i) - \sum_i^n (q_i \log q_i - q_i) - \langle \boldsymbol{p} - \boldsymbol{q}, \nabla_{\mathrm{ri}} \phi(\boldsymbol{q}) \rangle_{\mathcal{A}} \\
&= \sum_i^n (p_i \log p_i) - \sum_i^n (q_i \log q_i) + \cancelto{0}{\sum_i (q_i - p_i)} - \\
&\left\langle \boldsymbol{p} - \mathbf{1}, \log q_i - \frac{1}{n} \sum_i^n \log q_i \right\rangle + \left\langle \boldsymbol{q} - \mathbf{1}, \log q_i - \frac{1}{n} \sum_i^n \log q_i \right\rangle \\
&= \sum_i^n p_i \log \left(\frac{p_i}{q_i} \right) + \cancelto{0}{\left\langle (\frac{1}{n} \sum_i^n \log q_i) \mathbf{1}, (\boldsymbol{p} - \boldsymbol{q}) \right\rangle} + \\
&\quad \cancelto{0}{\sum_i^n (q_i \log q_i) - \sum_i^n (q_i \log q_i)} \\
&= \mathrm{KL}\left(\boldsymbol{p} \| \boldsymbol{q}\right).
\end{aligned}
\tag{2.8}
$$

In the equality (a) we have used equation (2.5)*.*

Note that definition (2) subsumes definition (1). To minimize clutter of notation we will not decorate the inner product and the relative gradient specifically. Whether the dot-product used is relative used will be evident from context (essentially from the nature of the interior of the domain of the function ϕ used to generate the Bregman divergence).

Example 2. *Consider the following function with domain* ▲

$$
\phi(\boldsymbol{p}) = \begin{cases} \sum_i (p_i \log p_i) + (1 - \sum_i p_i) \log(1 - \sum_i p_i) & \text{for } \boldsymbol{p} \in \blacktriangle \subset \mathbb{R}^{n-1} \\ +\infty & \text{otherwise}. \end{cases}
\tag{2.9}
$$

The term $(1 - \sum_i p_i) \log(1 - \sum_i p_i)$ *is closed and strictly convex function of* p_i *because it*

is an affine precomposition of a closed and convex function $x \log x$*. The gradient of* (2.9) *is*

$$\nabla\phi(\boldsymbol{p}) = \begin{bmatrix} \vdots \\ \log\left(\frac{p_i}{1-\sum_i p_i}\right) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n-1}.$$

One can verify that the Bregman divergence obtain from (2.9) *has the same form as KL divergence but defined as a mapping* $(\blacktriangle, \blacktriangle) \mapsto \mathbb{R}+$*. Furthermore, unlike* (2.6) *the function* (2.9) *is a Legendre function with a non-empty interior* $\operatorname{int} \blacktriangle$*. As a result there is an one to one correspondence with the domain of* ϕ *and its Legendre conjugate* ϕ^* *via the mapping* $\nabla\phi$ *and* $(\nabla)^{-1}\phi$*.*

2.2.1 Bregman Projection

One can define a projection operation in terms of Bregman divergences. Given a closed set $\mathcal{S}$, the Bregman-projection of $\boldsymbol{q}$ on $\mathcal{S}$ is $\operatorname{Proj}^{\phi}(q, \mathcal{S}) \triangleq \operatorname{Argmin}_{\boldsymbol{p}} D_{\phi}\big(\boldsymbol{p}\big|\big|\boldsymbol{q}\big) \quad \boldsymbol{p} \in \mathcal{S}$. A result (lemma 1) similar to Pythagoras theorem holds for the projection $\operatorname{Proj}^{\phi}(q, \mathcal{S})$ of a point $\boldsymbol{p}$ outside the convex set $\mathcal{S}$ on $\mathcal{S}$. One can show that for the same point $\boldsymbol{p}$, its projection on the supporting hyperplane of $\mathcal{S}$ passing through the projection $\operatorname{Proj}^{\phi}(q, \mathcal{S})$ coincides with it. This result allows us to reduce the case of projection on convex sets to projections on suitable hyperplanes.

Lemma 1. *(Censor and Lent, 1981) Consider the Bregman projection* $\operatorname{Proj}^{\phi}(q, \mathcal{S})$ *of* $\boldsymbol{q}$ *on a convex set* $\mathcal{S}$ *and the supporting hyperplane* $\mathcal{H} = \{\boldsymbol{x} | \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b\}$ *of the convex set* $\mathcal{S}$ *through* $\operatorname{Proj}^{\phi}(q, \mathcal{S})$ *. Then*

$$D_{\phi}\big(\boldsymbol{x}\big|\big|\boldsymbol{q}\big) = D_{\phi}\big(\boldsymbol{x}\big|\big|\operatorname{Proj}^{\phi}(q, \mathcal{S})\big) + D_{\phi}\big(\operatorname{Proj}^{\phi}(q, \mathcal{S})\big|\big|\boldsymbol{q}\big) \quad \forall_{\boldsymbol{x} \in \mathcal{H}}.$$

Lemma 2. *(Censor and Lent, 1981) Given a hyperplane $\mathcal{H}_1 = \{\boldsymbol{x} | \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b_1\}$, the Bregman projection $\mathrm{Proj}^{\phi}(q, \mathcal{H}_1)$ satisfies the equation*

$$\nabla\phi(\mathrm{Proj}^{\phi}(q, \mathcal{H}_1)) = \nabla\phi(q) + \lambda(\mathcal{H}_1)\boldsymbol{a},$$

for some $\lambda(\mathcal{H}_1)$ and the symmetrized Bregman divergence between $\boldsymbol{q}$ and its projection is given by

$$D_{\phi}\big(\mathrm{Proj}^{\phi}(q, \mathcal{H}_1)\big|\big|\boldsymbol{q}\big) + D_{\phi}\big(\boldsymbol{q}\big|\big|\mathrm{Proj}^{\phi}(q, \mathcal{H}_1)\big) = \lambda(\mathcal{H})(b - \langle \boldsymbol{a}, \boldsymbol{q} \rangle).$$

For a parallel hyperplane $\mathcal{H}_2 = \{\boldsymbol{x} | \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b_2\}$ with $b_2 \geq b_1$, we have $\lambda(\mathcal{H}_2) \geq \lambda(\mathcal{H}_1)$. Consider any point $\boldsymbol{y}$ such that $\mathcal{H}_1$ lies between $\boldsymbol{y}$ and $\mathcal{H}_2$, then

$$D_{\phi}\big(\mathrm{Proj}^{\phi}(\boldsymbol{y}, \mathcal{H}_2)\big|\big|\boldsymbol{y}\big) = D_{\phi}\big(\mathrm{Proj}^{\phi}(\boldsymbol{y}, \mathcal{H}_2)\big|\big|\boldsymbol{y}\big) + D_{\phi}\big(\mathrm{Proj}^{\phi}(\boldsymbol{y}, \mathcal{H}_2)\big|\big|\mathrm{Proj}^{\phi}(\boldsymbol{y}, \mathcal{H}_1)\big).$$

2.2.2 Exponential Families, Generalized Linear Models and Bregman Divergences

Bregman proposed the family of Bregman divergences as a means of solving convex optimization problems. Perhaps surprisingly, these divergences are fundamentally related to exponential family distributions. Their intimate connection plays an important role in this dissertation. A brief review follows:

A natural exponential family *density* [2] of a random variable Y has the form

$$P(Y = \boldsymbol{y} \mid \boldsymbol{\theta}) = \exp^{\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - \psi(\boldsymbol{\theta})}.$$

These densities are indexed by what is known as its *natural* parameter $\boldsymbol{\theta}$. It is well known

[2] with respect to a base measure. For notational simplicity the base measure will be dropped.

(Lehmann, 1983) that not only is the domain

$$\Theta = \left\{ \boldsymbol{\theta} \,\middle|\, \int_{\mathcal{Y}} \exp^{\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle} < \infty \right\}$$

of the parameter a convex set, the normalizer $\psi(\boldsymbol{\theta})$, a function defined on Θ, is also a convex function (strictly so if $\mathcal{Y}$ is affinely independent). Also called the log partition function, $\psi(\boldsymbol{\theta})$ is of great importance because all moments of Y can be recovered from it, for example

$$\mathbb{E}\left[Y\right] = \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}).$$

In statistics and machine learning one is interested in an estimate of the parameter $\boldsymbol{\theta}$ that generated a sample $\boldsymbol{y}$. Maximum likelihood obtains such an estimate $\boldsymbol{\theta}^*$ as the maximizer of the sample log likelihood, or equivalently as the solution of the following optimization problem

$$\begin{aligned}
\boldsymbol{\theta}^* &= \operatorname{Argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{y} \mid \boldsymbol{\theta}) \\
&= \operatorname{Argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{y} \mid \boldsymbol{\theta}) - \log P(\boldsymbol{y} \mid \boldsymbol{\theta}^*) = \operatorname{Argmin}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}^*) - \langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \boldsymbol{y} \rangle \\
&= \operatorname{Argmin}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}^*) - \langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}^*) \rangle \quad \text{[using optimality of } \boldsymbol{\theta}^*] \\
&= \operatorname{Argmin}_{\boldsymbol{\theta}} D_{\psi}(\boldsymbol{\theta} || \boldsymbol{\theta}^*) = \operatorname{Argmin}_{\boldsymbol{\theta}} D_{\boldsymbol{\phi}}\big(\boldsymbol{y} \,\big|\big|\, (\nabla \phi)^{-1}(\boldsymbol{\theta})\big) \quad \text{[using (2.4)]}
\end{aligned} \tag{2.10}$$

Generalized linear models (GLM) assume an exponential family probability density for Y conditioned on observed features $\boldsymbol{x}$. The parameter $\boldsymbol{\theta}$ is assumed to be a linear function of $\boldsymbol{x}$, as a result the corresponding conditional maximum likelihood optimization problem is

$$\boldsymbol{\theta}^* = \operatorname{Argmin}_{\boldsymbol{\theta}} D_{\boldsymbol{\phi}}\big(\boldsymbol{y} \,\big|\big|\, (\nabla \phi)^{-1}(\langle \boldsymbol{x}, \boldsymbol{w} \rangle)\big).$$

Bregman's algorithm:

Initialize: $\boldsymbol{\lambda}^0 \in \mathbb{R}+^d$ and z^0 such that
$\nabla\phi(z^0) = \left[\boldsymbol{A}^\dagger | \nabla\phi(\boldsymbol{y})\right]\left[\boldsymbol{\lambda}^{0\dagger}, 1\right]^\dagger$

Repeat: Till convergence

Update: Apply **Sequential** or **Parallel Update** to obtain λ^{t+1}

Solve: $\nabla\phi(z^{t+1}) = \left[\boldsymbol{A}^\dagger | \nabla\phi(\boldsymbol{y})\right]\left[\boldsymbol{\lambda}^{t+1\dagger}, 1\right]^\dagger$

Sequential Bregman Update:

Select i**:** Let $\mathcal{H}_i = \{\boldsymbol{z} | \langle \boldsymbol{a}_i, \boldsymbol{z}\rangle \leq b_i\}$

Compute $\mathrm{Proj}^\phi\left(\boldsymbol{z}^t, \mathcal{H}_i\right), c_i^t$ (see Lemma 2)
$\nabla\phi(\mathrm{Proj}^\phi\left(\boldsymbol{z}^t, \mathcal{H}_i\right)) = \nabla\phi(\boldsymbol{z}^t) + c_i^t \boldsymbol{a}_i,$

Update: $\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + c_i^t \mathbf{1}_i$

Parallel Bregman Update:

For all i **in parallel:** Compute
$\mathrm{Proj}^\phi\left(\boldsymbol{z}^t, \mathcal{H}_i\right), c_i^t$, (Lemma 2)

Update: $\boldsymbol{\lambda}_i^{t+1} = \boldsymbol{\lambda}^t + c_i^t \mathbf{1}_i$

Synchronize: $\boldsymbol{\lambda}^{t+1} = (\nabla\phi)^{-1}(\sum_i \nabla\phi(\boldsymbol{\lambda_i}^{t+1}))$

Table 2.1: Bregman's Algorithm

2.2.3 Bregman's Algorithm

Bregman divergences were first proposed (Bregman, 1967) in the context of a generalization of alternating orthogonal projection based algorithm for solving convex optimization problems, in particular

$$\min_{\boldsymbol{x}} D_\phi\left(\boldsymbol{x} \middle|\middle| \boldsymbol{y}\right) \text{ s.t. } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}. \tag{2.11}$$

A significant advantage of Bregman's algorithm is its scalability and suitability for parallelization. The algorithm operates by repeatedly projecting a dual feasible point onto the constraints using Bregman projections. We list the algorithm in Table 2.1. Readers may make special note of the simplicity of the parallel variant which applies directly to MR. This ease of parallelization was one the many reasons for basing the MR framework on Bregman divergences.

Chapter 3

Monotone Retargeting

This chapter introduces a novel approach for learning to rank (LETOR) based on the notion of monotone retargeting. Monotone retargeting minimizes a divergence between all monotonic increasing transformations of the relevance scores and a parameterized prediction function. The minimization is over the transformations as well as over the parameters. MR is applied with Bregman divergences, a large class of "distance like" functions that were recently shown to be the unique class that is statistically consistent with the normalized discounted gain (NDCG) criterion (Ravikumar et al., 2011). The algorithm uses alternating projection style updates, in which one set of simultaneous projections can be computed independent of the Bregman divergence and the other reduces to parameter estimation of a generalized linear model. This results in an easily implemented, efficiently parallelizable algorithm for the LETOR task that enjoys global optimum guarantees under mild conditions. We present empirical results on benchmark datasets showing that this approach can substantially outperform the state of the art NDCG consistent techniques.

This chapter is organized as follows: In Section 3.1 we present a reduction of an optimization problem over the infinite class of all monotonic increasing functions to that of alternating projection over a finite dimensional vector space. We introduce Bregman divergences in Section 3.2 and discuss properties that make them particularly suited to the

ranking task. We show (i) that one set of the alternating projections can be computed in a Bregman divergence independent fashion (in Section 3.2.1), and (ii) separable Bregman divergences allow us to use sorting (in Section 3.2.2) that would have otherwise required exhaustive combinatorial enumeration or solving a linear assignment problem repeatedly. In Section 3.2.3 we show when that optimization problem is jointly convex by resolving the question of joint convexity of the Fenchel-Young gap.

Notation: Vectors are denoted by bold lower case letters, matrices are capitalized. $\boldsymbol{x}^\dagger$ denotes the transpose of the vector $\boldsymbol{x}$, $||\boldsymbol{x}||$ denotes the L_2 norm. $\mathrm{Diag}(\boldsymbol{x})$ denotes a diagonal matrix with its diagonal set to the vector $\boldsymbol{x}$. $\text{Adj-Diff}(\boldsymbol{x})$ denotes a vector obtained by taking adjacent difference of consecutive components of $\left[\begin{smallmatrix}\boldsymbol{x}\\0\end{smallmatrix}\right]$, thus $\text{Cum-Sum}(\text{Adj-Diff}(\boldsymbol{x})) = \boldsymbol{x}$. A vector $\boldsymbol{x}$ is defined to be in *descending order* if $x_i \geq x_j$ when $i > j$, the set of such vectors is denoted by $\mathcal{R}\!\downarrow$. Vector $\boldsymbol{x}$ is isotonic with $\boldsymbol{y}$ if $x_i \geq x_j$ implies $y_i \geq y_j$. The unit simplex is denoted by Δ and the positive orthant by ${\mathbb{R}_+}^d$. Everywhere the symbol $\psi(\cdot)$ appears in this chapter it is used to denote the Legendre dual of the function $\phi(\cdot)$.

Background: In the chapter we make heavy use of known identities and algorithms associated with Bregman divergences and their relation to generalized linear models and exponential family distributions. Chapter 2 summarizes the necessary background. Several new properties of Bregman divergences particularly relevant to the LETOR problem are described in Section 3.2

Structured output space models (Bakir et al., 2007) have dominated the task of learning to rank (LETOR). Point-wise regression based models (introduced in Chapter 1) have been superseded by pairwise models (Freund et al., 2003), which in turn are being gradually displaced by list-wise approaches (Cao et al., 2007b; Lan et al., 2009). This trend has on one hand greatly improved the quality of the predictions obtained but on the other hand has come at the cost of additional complexity and computation. The cost functions of structured models are often defined directly on the combinatorial space of permutations,

which significantly increase the difficulty of learning and optimization compared to regression based approaches. We propose an approach to the LETOR task that retains the simplicity of the regression based models, is simple to implement, is embarrassingly parallelizable, and yet is a function of ordering alone. Furthermore, the resulting algorithm enjoys strong guarantees of convergence, statistical consistency under uncertainty and a global minimum under mild conditions. Our experiments on benchmark datasets show that the proposed approach outperforms state of the art models in terms of several common LETOR metrics.

We adapt regression to the LETOR task by using monotone retargeting (MR) and Bregman divergences. MR is a novel technique that we introduce in this chapter and Bregman divergences (Bregman, 1967) are a family of "distance like" functions well studied in optimization (Censor and Lent, 1981), statistics and machine learning (Banerjee et al., 2005) (See Chapter 1 for details). Bregman divergences are also the unique class of strongly statistically consistent surrogate cost functions for the NDCG criterion (Ravikumar et al., 2011), a de facto standard of ranking quality. In addition to these statistical characteristics, Bregman divergences have several properties useful for optimization. and as we shall show, specifically useful for ranking.

By combining Bregman divergences and MR we obtain provably convergent coordinate descent algorithms with guarantees of global minimum under conditions easy to satisfy. The LETOR task decomposes into subproblems that are equivalent to estimating (unconstrained as well as constrained) generalized linear models. The Bregman divergence machinery provides easy to implement, scalable algorithms for them, with a user chosen level of granularity of parallelism. We hope the reader will appreciate the flexibility of choosing an appropriate divergence to encode desirable properties on the rankings while enjoying the strong guarantees.

We motivate MR by first discussing direct regression of rank scores and highlighting its primary deficiency: its attempt to fit the scores exactly. An exact fit is unnecessary since any score that induces the correct ordering is sufficient. MR addresses this problem by

searching for a order preserving transformation of the target scores that may be easier for the regressor to fit: hence the name "retargeting". Searching over all monotonic transformations is a unique characteristic of MR.

3.1 Monotone Retargeting

Consider a set of queries $\mathcal{Q} = \{q_1, q_i \dots q_{|\mathcal{Q}|}\}$ and a set of items $\mathcal{V}$ that are to be ranked in the context of the queries. For every query q_i there is a subset $\mathcal{V}_i \subset \mathcal{V}$ whose elements have been ordered, based on their relevance. This ordering is customarily expressed via a rank score vector $\tilde{\boldsymbol{r}}_i \in \mathbb{R}^{d_i=|\mathcal{V}_i|}$ whose components $\tilde{r}_{ij}$ correspond to items in $\mathcal{V}_i$. In this chaper we assume that beyond establishing an order over the set $\mathcal{V}_i$, the actual values of $\tilde{r}_{ij}$ are of no significance. For a query q_i the index j of $\tilde{r}_{ij}$ is local to the set $\mathcal{V}_i$ hence $\tilde{r}_{ij}$ and $\tilde{r}_{kj}$ need not correspond to the same object. We shall further assume, with no loss in generality, that the subscript j is assigned such that $\tilde{r}_{ij}$ is in a descending order for any $\mathcal{V}_i$. Note that $\tilde{\boldsymbol{r}}_i$ induces a partial order if the number of unique values k_i in the vector is less than d_i. For every query-object pair $\{q_i, v_{ij}\}$ a feature vector $\mathbb{R}^n \ni \boldsymbol{a}_{ij} = F(q_i, v_{ij})$ is computed apriori with some predefined F. The subset of training data pertinent to any query q_i is the pair $\{\tilde{\boldsymbol{r}}_i, \boldsymbol{A}_i\}$ and is called its qset. The column vector $\tilde{\boldsymbol{r}}_i$ consists of the rank-scores $\tilde{r}_{ij}$ and $\boldsymbol{A}_i$ is a matrix whose j^{th} row is $\boldsymbol{a}_{ij}{}^{\dagger}$.

Given a loss function $D_i : \mathbb{R}^{\boldsymbol{s}} \times \mathbb{R}^{\boldsymbol{s}} \mapsto \mathbb{R}_+$ we may define a regression model $\min_{\boldsymbol{w}} \sum_i D(\tilde{\boldsymbol{r}}_i, f(\boldsymbol{A}_i, \boldsymbol{w}))$ where $f : \mathbb{R}^{\boldsymbol{s}\times n} \times \mathbb{R}^n \mapsto \mathbb{R}^{\boldsymbol{s}}$ is some fixed parametric form with the parameter $\boldsymbol{w}$. This is a common approach and in the context of LETOR these are called point-wise methods. As discussed, this is unnecessarily stringent for ranking. A better alternative is:

$$\min_{\boldsymbol{w}, \boldsymbol{\Upsilon}_i \in \mathcal{M}} \sum_i D_i\big(\tilde{\boldsymbol{r}}_i, \boldsymbol{\Upsilon}_i \circ f(\boldsymbol{A}_i, \boldsymbol{w})\big),$$

where $\boldsymbol{\Upsilon}_i : \mathbb{R}^{\boldsymbol{s}} \mapsto \mathbb{R}^{\boldsymbol{s}}$ transforms the component of its argument by a fixed monotonic, strictly increasing function Υ_i, and $\mathcal{M}$ is the class of all such functions. Now $f(\boldsymbol{A}_i, \boldsymbol{w})$ no

longer need to equal $\tilde{\boldsymbol{r}}_i$ point-wise to incur zero loss. It is sufficient for some monotonic increasing transform of $f(\boldsymbol{A}_i, \boldsymbol{w})$ to do so.

Optimizing a suitable loss function over all possible monotonic, strictly increasing function Υ_i is the topic of Chapter 4. In this chapter we take simpler route of applying the monotonic transform to $\tilde{\boldsymbol{r}}_i$ and optimize over the range space generated. This avoids the minimization over the function composition, but the need for minimizing over the range space of all monotone functions remains. One possible way to eliminate the minimization over the function space is to restrict our attention to some parametric family in $\mathcal{M}$ at the expense of generality. Instead, with no loss in generality, the optimization over the infinite space of functions $\mathcal{M}$ can be converted into one over finite dimensional vector spaces $\mathbb{R}^{|\mathcal{V}_j|}$, provided we have a finite characterization of the constraint set $\mathcal{R}{\downarrow_i}$ defined as below:

$$\min_{\boldsymbol{w}, \boldsymbol{r} \in \mathcal{R}{\downarrow_i}} \sum_i D_i(\boldsymbol{r}_i, f(\boldsymbol{A}_i, \boldsymbol{w})) \text{ s.t. } \mathcal{R}{\downarrow_i} = \left\{ \boldsymbol{r} | \, {}^{\exists \boldsymbol{M} \in \mathcal{M}}_{\boldsymbol{M}(\tilde{\boldsymbol{r}}_i) = \boldsymbol{r}} \right\}. \tag{3.1}$$

The Set $\mathcal{R}{\downarrow_i}$: It is the set of vectors isotonic to $\tilde{\boldsymbol{r}}_i$. The convex composition $\boldsymbol{r} = \alpha \boldsymbol{r}_1 + (1 - \alpha)\boldsymbol{r}_2$ of two isotonic vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ preserves isotonicity, as does the scaling $\alpha \boldsymbol{r}_1$ for any $\alpha \in \mathbb{R}_+$. Hence the set $\mathcal{R}{\downarrow_i}$ is a convex cone. This makes the problem computationally tractable because the set can be described entirely by its extreme rays, or by the extreme rays of its polar. We claim the set $\mathcal{R}{\downarrow_i}$ can be expressed as the image of the set $\{\mathbb{R}_+\}^{\boldsymbol{s}-1} \times \mathbb{R}$ under a linear transformation by a particular upper triangular matrix U with positive entries:

$$\mathcal{R}{\downarrow_i} = U\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x} \in \{\mathbb{R}_+\}^{\boldsymbol{s}-1} \times \mathbb{R}$$

The matrix U is not unique and can be generated from any vector $\boldsymbol{v} \in {\mathbb{R}_+}^{\boldsymbol{s}}$, but as we shall see, any member from the allowed class of U is sufficient for an *exhaustive* representation of $\mathcal{R}{\downarrow_i}$. [1]

[1] For regression functions capable of fitting an arbitrary additive offset, no generality is lost by constraining the last component of $\boldsymbol{x}$ to be non-negative.

Lemma 3. *The set of all vectors in $\mathbb{R}^d$ that are sorted in a descending order is given by $U\boldsymbol{x}$ s.t. $\boldsymbol{x} \in \{\mathbb{R}_+\}^{s-1} \times \mathbb{R}$ where U is a triangular matrix generated from a vector $\boldsymbol{v} \in {\mathbb{R}_+}^d$ such that the i^{th} row $U(i,:)$ is $\{0\}^{i-1} \times \boldsymbol{v}(i:)$*

Proof. Consider solving $U\boldsymbol{x} = \tilde{\boldsymbol{r}}_i$ for any vector $\tilde{\boldsymbol{r}}_i$ sorted in descending order. We have $\boldsymbol{x} = (\text{Diag})^{-1}(\boldsymbol{v}) \times \text{Adj-Diff}(\tilde{\boldsymbol{r}}_i)$ which is in $\{\mathbb{R}_+\}^{s-1} \times \mathbb{R}$ □

The Set Δ_o^i: In addition to the set $\mathcal{R}{\downarrow_i}$ we shall make frequent use of the set of all discrete probability distributions that are in descending order, i.e. $\mathcal{R}{\downarrow_i} \cap \Delta_i$ that we represent by Δ_o^i. The choice of this set is motivated by two reasons, to keep the contribution of different qsets comparable in the cost function, and the need to keep the rank-score vector bounded away from the origin. Similar to the set $\mathcal{R}{\downarrow}$, we may represent this set by generating an upper triangular matrix T from the vector $\boldsymbol{v}_\Delta = \{1, \frac{1}{2}, \cdots \frac{1}{i} \cdots \frac{1}{d}\}$ and considering $\boldsymbol{x} \in \Delta$.

Lemma 4. *The set Δ_o of all discrete probability distributions of dimension d that are in descending order is the image $T\boldsymbol{x}$ s.t. $\boldsymbol{x} \in \Delta$ where T is an upper triangular matrix generated from the vector $\boldsymbol{v}_\Delta = \{1, \frac{1}{2} \cdots \frac{1}{d}\}$ such that $T(i,:) = \{0\}^{i-1} \times \boldsymbol{v}_\Delta(i:)$*

Proof. The proof follows Lemma (3). $T\boldsymbol{x}$ is in the simplex Δ because it is a convex combination of vectors in Δ. □

Given any choice of the distance like function $D_i(\cdot,\cdot)$ and the curve fitting function $f(\cdot,\cdot)$ we obtain an optimization problem that can be optimized alternately in the rank scores and parameters of f. It will certainly be convenient if the resulting optimization problem is convex. We show that (i) by choosing $D_i(\cdot,\cdot)$ to be a Bregman divergence $D_\phi\big(\cdot\big|\big|\cdot\big)$ obtained from a convex function $\phi(\cdot)$ and (ii) $f(\cdot,\cdot)$ to be a matching curve fitting function $(\nabla\phi)^{-1}({\boldsymbol{A}_i}^\dagger \boldsymbol{w})$, one obtains from (3.1) a bi-convex optimization[2] problem over a

[2]A biconvex function is a function of two arguments such that with any one of its arguments fixed the function is convex in the other argument.

Function: $\boldsymbol{\phi}(\boldsymbol{x})$	Divergence: $D_\phi\big(\boldsymbol{x}\big\|\big\|\boldsymbol{y}\big)$	Link: $(\nabla\phi)^{-1}(\boldsymbol{x})$
$\frac{1}{2}\|\|\boldsymbol{x}\|\|_W^2$	$\frac{1}{2}\|\|\boldsymbol{x}-\boldsymbol{y}\|\|_W^2$	$\boldsymbol{x}$
$\sum_i w_i x_i \log x_i \ \boldsymbol{x}\in\Delta$	$wKL\left(\boldsymbol{x}\|\boldsymbol{y}\right)=\sum_i w_i x_i \log(\frac{x_i}{y_i})$	$\frac{\exp(\boldsymbol{x})}{\sum_i \exp(x_i)}$
$\sum_i w_i(x_i\log x_i - x_i)$ $\boldsymbol{x}\in {\mathbb{R}_+}^d$	$wGI\left(\boldsymbol{x}\|\boldsymbol{y}\right)$ $=\sum_i w_i\big((x_i-1)\log(\frac{x_i-1}{y_i-1})-x_i+y_i\big)$	$\exp(\boldsymbol{x})$

Table 3.1: Examples of WIS Bregman divergences.

product of convex sets.

$$\min_{\boldsymbol{w},\boldsymbol{r}\in\mathcal{R}\downarrow_i}\sum_{i=1}^{|\mathcal{Q}|}\frac{1}{|\mathcal{V}_i|}D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i\big\|\big|(\nabla\phi)^{-1}\left(\boldsymbol{A}_i{}^\dagger\boldsymbol{w}\right)\big). \tag{3.2}$$

Readers familiar with GLMs will recognize that optimization with respect to $\boldsymbol{w}$ in (3.2) is nothing but maximum log likelihood estimation of a GLM with the canonical link function $(\nabla\phi)^{-1}(\cdot)$, as discussed briefly in Section 2.2.2 (see equation (2.10)). Table 3.1 shows some common Bregman divergences, the convex functions generating them and their corresponding link functions. The optimization with respect to $\boldsymbol{r}\in\mathcal{R}\downarrow_i$ can also be seen as maximum log likelihood estimation of an exponential family, but under linear constraints on the parameters, for which scalable techniques are available, (see (2.2.3), (Censor, 1981)). The LETOR task has additional structure in the type of linear constraints imposed and these can be exploited to give efficient solutions, as we shall see shortly. In the actual LETOR task we augment (3.2) with a convex regularization term to take care of overfitting.

3.2 Ranking Related Properties

In this section we explore properties that make the Bregman divergence based cost function (3.2) particularly suitable for learning ranking. We shall see that the minimization over $\boldsymbol{r}$ can be made (almost) agnostic to the function $\phi(\cdot)$. The use of separable Bregman divergences also allows one to obtain the best re-permutation of $\boldsymbol{r}$ that minimizes the cost function where all other terms stay constant. Finally, we show under what conditions the

cost function is not only separately convex in $\boldsymbol{r}$ and $\boldsymbol{w}$, which is always guaranteed, but also jointly convex. Although these properties play a pivotal role in the monotone retargeting formulation they are also significant in their own right.

3.2.1 Universality of Minimizers over Ordered Sets

A mean-variance like decomposition (described in appendix A.1, Theorem (15)) holds for all Bregman divergences. It plays a critical role in Theorem 1 which has significant impact in facilitating the solution of the LETOR problem.

Proposition 1. *For $\mathcal{R}\!\downarrow \subset \mathbb{R}^d$ the entire set of vectors with descending ordered components, the minimizer $\boldsymbol{y}^* = \underset{\boldsymbol{y}\in\mathcal{R}\downarrow}{\operatorname{Argmin}} D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big)$ is independent of $\boldsymbol{\phi}(\cdot)$ if $\boldsymbol{\phi}(\cdot)$ is WIS.*

Proof. A more general case is proven in Proposition 2 □

Following our independent proof of Proposition 1, we have since come across an older proof (Barlow and Brunk, 1972) developed prior to the popularity of Bregman divergences and in the context of maximum likelihood estimators of exponential family models under conic constraints. Whereas the older proof uses Moreau's cone decomposition (Rockafellar, 1996), ours uses Theorem 15 (in appendix A) and yields a much shorter proof.

Corollary 1. *If $\operatorname{dom}\psi(\cdot) = \mathbb{R}^d$ where $\psi(\cdot)$ is the Legendre conjugate of the WIS convex function $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{z}^* = \operatorname{Argmin}_{\boldsymbol{z}\in\mathcal{R}\downarrow} ||\boldsymbol{x}-\boldsymbol{z}||^2$ then*

$$\underset{\boldsymbol{y}\in\mathcal{R}\downarrow\cap\operatorname{dom}\phi}{\operatorname{Argmin}} D_\phi\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{x})\big) = (\nabla\boldsymbol{\phi})^{-1}(\boldsymbol{z}^*).$$

Note that Corollary 1 is directly applicable to formulation (3.2). It implies that for an infinitely large class of convex functions $\boldsymbol{\phi}(\cdot)$ for which the dual domain is $\mathbb{R}^d$, the minimization over $\boldsymbol{r}_i \in \mathcal{R}\!\downarrow\cap\operatorname{dom}\boldsymbol{\phi}$ can be obtained by transforming the equivalent squared loss minimizer by $(\nabla\boldsymbol{\phi})^{-1}(\cdot)$. The squared loss minimization is not only simpler but its source code implementation can now be shared across instantiations of (3.2) with different

$\boldsymbol{\phi}(\cdot)$s whenever Corollary 1 applies. It is clear from the precondition of the corollary that the class of convex functions where the corollary applies is identical to those defined as "essentially smooth" (Rockafellar, 1996). Three such functions are listed in Table 3.1.

3.2.2 Optimality of Sorting

For any sorted vector $\boldsymbol{x}$, finding the permutation of $\boldsymbol{y}$ that minimizes $D_\phi(\boldsymbol{x} \Big|\Big| \boldsymbol{y})$ shows up as a subproblem in our formulation that needs to be solved in an inner loop. Thus solving it efficiently is critical and this is yet another instance where Bregman divergences are very useful.

For an arbitrary divergence function the search for the optimal permutation is a *non-linear assignment* problem that can be solved only by exhaustive enumeration. For an arbitrary separable divergence the optimal permutation may be found by solving a linear assignment problem, which is an integer linear program and expensive to solve (especially in an inner loop, as required in our algorithm). On the other hand, if $\boldsymbol{\phi}(\cdot)$ is IS, the solution is remarkably simple, as shown in Lemma 5 where $\left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right]$ denotes a partitoned vector with vector components x_1 and x_2.

Lemma 5. *If $x_1 \geq x_2$ and $y_1 \geq y_2$ and $\boldsymbol{\phi}(\cdot)$ is IS, then $D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{smallmatrix}\right]\big) \leq D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{y}_2\\ \boldsymbol{y}_1\end{smallmatrix}\right]\big)$ and $D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right]\big) \leq D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{y}_2\\ \boldsymbol{y}_1\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right]\big)$.*

Proof. $D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{smallmatrix}\right]\big) - D_\phi\big(\left[\begin{smallmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{smallmatrix}\right] \Big|\Big| \left[\begin{smallmatrix}\boldsymbol{y}_2\\ \boldsymbol{y}_1\end{smallmatrix}\right]\big) = \langle(\nabla\phi(y_2) - \nabla\phi(y_1)), x_1 - x_2\rangle$. There exists $c \geq 0$ s.t. $x_1 - x_2 = c(y_1 - y_2)$. Proof follows from monotonicity of $\nabla\phi$, ensured by convexity of ϕ. We can exchange the order of the arguments using the property (2.4). □

Using induction over d for $\boldsymbol{y} \in \mathbb{R}^d$ the optimal permutation is obtained by sorting. Not only is Lemma 5 extremely helpful in generating descent updates, it has fundamental consequences related to the local and global optimum of our formulation (see Lemma 6).

3.2.3 Joint Convexity and Global Minimum

In this section we are concerned about the joint convexity of the formulation (3.2). Joint convexity, if ensured, guarantees global minimum even for coordinate-wise minimization because the objective function is smooth and the constraint set is a Cartesian product of convex sets.

Using Legendre duality one recognizes that equation (3.2) quantifies the gap in the Fenchel-Young inequality (2.3) (normalized by $|\mathcal{V}_i|$).

$$D_{\boldsymbol{\phi}}\left(\boldsymbol{r}_i \middle|\middle| (\nabla\phi)^{-1}\left(\boldsymbol{A}_i\boldsymbol{w}\right)\right) = \psi(\boldsymbol{A}_i\boldsymbol{w}) + \phi(\boldsymbol{r}_i) - \langle \boldsymbol{r}_i, \boldsymbol{A}_i\boldsymbol{w}\rangle\,. \tag{3.3}$$

Although this establishes separate convexity in $\boldsymbol{w}$ and $\boldsymbol{r}_i$, the conditions under which joint convexity is obtained are not obvious. We resolve this important question in Theorem 1.

Theorem 1. *The gap in the Fenchel-Young inequality $\psi(\boldsymbol{y}) + \phi(\boldsymbol{x}) - \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ for any continuously differentiable, strictly convex $\phi(\cdot)$ with a differentiable conjugate $(\phi)^*(\cdot) = \psi(\cdot)$ is jointly convex if and only if, ignoring affine terms, $\phi(\boldsymbol{x}) = c||\boldsymbol{x}||^2$ for all $c > 0$.*

Proof: sketched in appendix A.

It follows from Theorem 1 that cost function 3.3 is jointly convex if and only if $\phi(\boldsymbol{x}) = c||\boldsymbol{x}||^2, c > 0$.

3.3 LETOR with Monotone Retargeting

Our cost function is an instantiation of (3.2) with a WIS Bregman divergence. In addition, we include regularization and a query specific offset. Note that the cost function (3.2) is not invariant to scale. Squared Euclidean, KL divergence and generalized I-divergence are homogeneous functions of degree 2, 1 and 1 respectively. Thus the cost can be reduced just by scaling its arguments down, without actually learning the task. To remedy this, we restrict the $\boldsymbol{r}_i$'s from shrinking below a pre-defined size. This is accomplished by constraining $\boldsymbol{r}_i$'s

to lie in an appropriate closed convex set separated from the origin, for example, an unit simplex or a shifted positive orthant. This yields:

$$\min_{\beta_i, \boldsymbol{w}, \boldsymbol{r}_i \in \mathcal{R}\downarrow_i \cap \mathcal{S}_i} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{|\mathcal{V}_i|} D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \Big|\Big| (\nabla\phi)^{-1}\left(\boldsymbol{A}_i \boldsymbol{w} + \beta_i \mathbf{1}\right)\big) + \frac{C}{2}||\boldsymbol{w}||^2, \tag{3.4}$$

or equivalently

$$\min_{\beta_i, \boldsymbol{w}, \boldsymbol{r}_i \in \mathcal{R}\downarrow_i \cap \mathcal{S}_i} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{|\mathcal{V}_i|} D_{\boldsymbol{\psi}}\big(\boldsymbol{A}_i \boldsymbol{w} + \beta_i \mathbf{1} \Big|\Big| \nabla\phi\left(\boldsymbol{r}_i\right)\big) + \frac{C}{2}||\boldsymbol{w}||^2, \tag{3.5}$$

where $\mathcal{S}_i$ are bounded sets excluding $\mathbf{0}$, chosen to suit the divergence. The parameter C is the regularization parameter. In non-transductive settings, the query specific offsets β_i will not be available for the test queries. This causes no difficulty because β_i does not affect the relative ranks over the documents. We update the $\boldsymbol{r}_i$'s and $\{\boldsymbol{w}, \{\beta_i\}\}$ alternately.

If $\mathcal{S}_i = \operatorname{dom}\boldsymbol{\phi}$ and $\operatorname{dom}\boldsymbol{\psi} = \mathbb{R}^d$, the optimization over $\boldsymbol{r}_i$ reduces to an order constrained least squares problem (corollary 1). Examples of such matched pairs are (i) $wKL\left(\cdot\|\cdot\right)$ and Δ_i, and (ii) shifted $wGI\left(\cdot\|\cdot\right)$ and $\mathbf{1} + {\mathbb{R}_+}^d$. A well studied, scalable algorithm for the ordered least squares problem is pool of adjacent violators (PAV) algorithm (Best and Chakravarti, 1990). One may also use Lemma 3 to solve it as a non-negative least squares problem for which several scalable algorithms exist (Kim et al., 2008).

To be able to use Bregman's algorithm, it is essential that $\mathcal{R}\downarrow_i$ be available as an intersection of linear constraints. This is readily obtained for any prescribed total order, as:

$$\begin{aligned} \mathcal{R}\downarrow_i &= \{r_{i,j+1} - r_{i,j} \leq 0\}_{\forall j \in \mathcal{J}_i}, \\ \Delta_i^o &= \mathcal{R}\downarrow_i \cap \{\sum_j r_{ij} = 1\} \cap \{r_{i,d_i} > 0\}. \end{aligned} \tag{3.6}$$

The advantages of the Bregman updates (2.2.3), are that they are easy to implement (more so when $\operatorname{Proj}^{\phi}\left(\cdot,\cdot\right)$ is available in closed form e.g. squared Euclidean) and have minimal memory requirements. Hence they scale readily and allow easy switch from a se-

quential to a parallel update. The parallel Bregman updates applied to (3.2), (3.6) clearly exposes massive amounts of fine grained parallelism at the level of individual inequalities in $\mathcal{R}\downarrow_i$ or Δ_i^o that can be exploited using Bregman's algorithm with parallel updates described in Section 2.2.3. They are well suited for implementation on a GPGPU (Nickolls et al., 2008). We note that the optimization for $\boldsymbol{r}_i$ is independent for each query, thus can be embarrassingly parallelized further. In our experiments on a representative set of largest available LETOR datasets (reported in Section 3.4) each iteration took no more than a couple of seconds, as a result we had little incentive for parallelization. However for industrial scale applications, for example ranking web pages, parallelization will play an important role.

For optimizing over $\boldsymbol{w}$ one may use several techniques available for parallelizing a sum of convex functions, for example, parallelizing the gradient computation across the terms or use more specialized technique such as alternating direction method of multipliers (Boyd et al., 2011). Further, $\{\boldsymbol{w}, \{\beta_i\}\}$ can be solved jointly simply by augmenting the feature matrix $\boldsymbol{A}_i$ with $\mathbf{1}$ for each query. We hope the readers will appreciate this flexibility of being able to exploit parallelism at different levels of granularity of choice.

3.3.1 Partial Order

Recall that a partial order is induced if the number of unique rank scores k_i in $\tilde{\boldsymbol{r}}_i$ is less than d_i. In this case, our convention of indexing $\mathcal{V}_i$ in a descending order is ambiguous. To resolve this, we break ties arbitrarily. Consider a subset of $\mathcal{V}_i$ whose elements have the same training rank-score. We distinguish between two modeling choices: (a) the items in that subset are not really equivalent, but the training set used a resolution that could not make fine distinctions between the items,[3] we call this the "hidden order" case, and (b) the items in the subset are indeed equivalent and the targets are constrained to reflect the same block structure, we call this case "block equivalent" and model it appropriately.

[3] or that, we only care to reduce the error of predicting $r_{ij} > r_{ik}$ when $\tilde{r}_{ij} < \tilde{r}_{ik}$. Note the strict inequality.

Partially Hidden Order

In this model we assume that the items are totally ordered, though the finer ordering between similar items is not visible to the ranking algorithm. Let $P_i = \{P_{ik}\}_{k=1}^{k_i}$ be a partition of the index set of $\mathcal{V}_i$, such that all items in P_{ik} have the same training rank-score. We denote their sizes by $d_{ik} = |P_{ik}|$. Although the relevance scores specify an order between items from two different sets P_{ij} and P_{il}, the order within any set P_{ik} remains unknown. The high cost of acquiring training data in a totally ordered form makes this scenario very common in practice.

The set $\mathfrak{R}_i$: Denote the set of rank-score vectors having the same partially ordered structure as $\tilde{\boldsymbol{r}}_i$ by $\mathfrak{R}_i$. For partial order we may describe $\mathfrak{R}_i$ by linear inequalities as follows:

$$\{r_{im} > r_{in}\}_{j=1}^{k_i-1} \forall_{i\in[1,|\mathcal{Q}|]},\ m \in P_{ij},\ n \in P_{i,j+1},$$

with each j generating $d_{ij}d_{i,j+1}$ inequalities. One may now replace the occurrence of $\mathcal{R}{\downarrow_i}$ by $\mathfrak{R}_i$ in the formulation 3.4 to obtain the formulation for the partial hidden order case. Thereby we obtain:

$$\min_{\beta_i,\boldsymbol{w},\boldsymbol{r}_i\in\mathfrak{R}_i\cap\mathcal{S}_i} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{|\mathcal{V}_i|} D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \Big|\Big| (\nabla\phi)^{-1}\left(\boldsymbol{A}_i\boldsymbol{w} + \beta_i\mathbf{1}\right)\big) + \frac{C}{2}||\boldsymbol{w}||^2. \tag{3.7}$$

The optimization problem may be solved using either an inner or an outer representation of the constraint sets, both offer different advantages.

Outer representation: Recall that Bregman's algorithm 2.2.3 is ideally suited for the outer representation (3.6). Note that the number of inequalities used in the representation of $\mathfrak{R}_i$ can be very large. This proliferation of inequalities may be controlled by introducing auxiliary variables $\{\bar{r}_{i,l}\}_{l=1}^{k_i-1}$ and inequalities:

$$\{\bar{r}_{i,j+1} > r_{iP_{ij}} > \bar{r}_{i,j}\}\forall_{i\in[1,|\mathcal{Q}|]}. \tag{3.8}$$

To this relatively parsimonious representation of $\mathfrak{R}_i$'s one may apply Bregman's algorithm to obtain the scores $\boldsymbol{r}_i$. However, since Bregman's algorithms are essentially coordinate-wise ascent methods, their convergence may be slow unless fine grained parallelism can be exploited, which are best performed in specialized hardware, for example GPGU (Nickolls et al., 2008). For commodity hardware, an alternative to the exterior point methods are proximal and interior point methods that use an inner representation of the convex constraint set. In our experiments we used the inner representation and proximal methods. Experimental details are in Section 3.4.

Inner representation: To construct the inner representation of the set of (hidden) partially ordered vectors we introduce a block-diagonally restricted permutation matrix $\mathbb{P}_i$ that, when multiplied to a vector, permutes the components in each P_{ij} independently. Since the items in P_{ij} are not equivalent they are available for re-ordering as long as that minimizes the cost (3.7). The inner representation of an arbitrary (hidden) partially ordered vector in $\mathbb{R}^n$ is therefore obtained as $\boldsymbol{r}_i = \mathbb{P}_i U \boldsymbol{x}_i$ with U and $\boldsymbol{x}$ as defined in lemma 3, and for ordered vectors in Δ_i, it is given by $\mathbb{P}_i T \boldsymbol{x}_i$, where Ti and $\boldsymbol{x}$ are as defined in lemma 4.

The cost function (3.7) may now be reduced by alternately minimizing over $\boldsymbol{x}_i, \mathbb{P}_i$ and $\boldsymbol{w}$. In our experiments we have used the inner representation and moreover we have constrained the score vectors to the simplex Δ_i to keep different qsets comparable and to keep the retargeted scores bounded away from $\mathbf{0}$ (see discussion preceding Lemma 4). The updates are shown in Figure 3.1.

When there are additional constraints on the set of hidden partially ordered score vectors, the vector $\boldsymbol{x}$ may be updated by the method of D proximal gradients, where the proximal term is a Bregman divergence defined by a Legendre convex function whose domain is the required constraint set (Iusem, 1997), (Censor and Zenios, 1992). We do not go into the details of proximal methods as it lies beyond the scope of this chapter, what is relevant is that this method automatically enforces the required constraints. Note that in the formulation (3.7) the additional constraint is denoted by $\mathcal{S}_i$. In our setting, the set $\mathcal{S}_i$ is Δ_i,

Input: Convex function ϕ, feature matrices $\{A_i\}$ with rows sorted by relevance, regularization parameter C.

Repeat Until Convergence:

$$\mathbb{P}_i^{t+1} = \underset{\pi}{\text{Argmin}}\, D_{\boldsymbol{\phi}}\big(T\boldsymbol{x}_i^t \Big|\Big| (\nabla\phi)^{-1}\left(\pi \boldsymbol{A}_i \boldsymbol{w}^t + \beta_i^t\right)\big) \;\; \forall i \tag{3.9}$$

$$\boldsymbol{x}_i^{t+1} = \underset{\boldsymbol{x}\in\Delta}{\text{Argmin}}\, D_{\boldsymbol{\phi}}\big(T\boldsymbol{x} \Big|\Big| (\nabla\phi)^{-1}\left(\mathbb{P}_i^{t+1} \boldsymbol{A}_i \boldsymbol{w}^t + \beta_i^t\right)\big) \;\; \forall i \tag{3.10}$$

$$\boldsymbol{w}^{t+1}, \{\beta_i^{t+1}\} = \underset{\boldsymbol{w},\{\beta_i\}}{\text{Argmin}} \sum_{i=1}^{|\mathcal{Q}|} D_{\boldsymbol{\phi}}\big(T\boldsymbol{x}_i^{t+1} \Big|\Big| (\nabla\phi)^{-1}\left(\mathbb{P}_i^{t+1} \boldsymbol{A}_i \boldsymbol{w} + \beta_i^t\right)\big) + \frac{C}{2}||\boldsymbol{w}||^2 \tag{3.11}$$

Return: $\boldsymbol{w}$.

Figure 3.1: Algorithm for Partially Hidden Order

in this case the corresponding proximal gradient update of $\boldsymbol{x}$ is the exponentiated gradient algorithm (Kivinen and Warmuth, 1995).

Recall that block weighted IS Bregman divergences have the special property that sorting minimizes the divergence over all permutations (Lemma 5). Thus update (3.9) can be accomplished by sorting. The $\mathbb{P}_i$ updates are obtained by sorting each block independently.

The updates (3.9), (3.10) and (3.11) each reduce the lower bounded cost (3.7), therefore the algorithm described in Figure 3.1 converges in function value. However, the vital question whether the updates converge to the stationary point of the cost function (3.7) remains. Make note of the fact that though (3.7) is differentiable in $\boldsymbol{r}_i$ it is not differentiable in the trifactored representation $\boldsymbol{r}_i = \mathbb{P}_i T \boldsymbol{x}_i$ because of the discrete nature of $\mathbb{P}_i$. The non differentiability may raise doubts about convergence to the stationary point of (3.7). Thus in the next couple of paragraphs we clarify why indeed the specified updates converge to such a stationary point.

Convergence to a Stationary Point: The tri-factored form $\boldsymbol{r}_i = \mathbb{P}_i U \boldsymbol{x}_i$ is a cause

for concern, though it is reassuring that the range of $\mathbb{P}_i U \boldsymbol{x}_i$ is $\mathfrak{R}_i$ which again is a convex cone and that the tri-factored representation of any point in that cone is described uniquely. This, however, is not sufficient to ensure that a minimum is achieved by (3.10) and (3.9) because though the constraint set is convex, the cost function (3.4) is not convex in the tri-factored parameterization. Worse still, the parameterization is discrete. [4]

If sorting (3.9) and constrained minimization (3.10) achieves the minimum $\boldsymbol{r}$ for a fixed $\boldsymbol{w}^{t+1}, \{\beta_i^{t+1}\}$, then convergence to the stationary point is guaranteed by the following theorem:

Theorem 2. *(Bertsekas, 1999) Let function $f(\boldsymbol{x}_1, \boldsymbol{x}_2)$ be continuously differentiable in its domain $\Pi\mathcal{X}_i$. Suppose for each i and $x \in \mathcal{X}_i$ the coordinate-wise minimum $\min_{\xi \in \mathcal{X}_i} f(\cdot, \xi, \cdot)$ is uniquely attained. Then every limit point of the sequence of coordinate-wise minimizers is a stationary point of f.*

Thus we explore the question whether (3.9) and (3.10) together achieve such a local minimum, because together they can be considered an instance of a meta-update that achieves minimality while the other parameters are kept fixed in a continuously differentiable cost function. Note that we may consider the permutation to be applied to the left argument without any loss of generality, because the divergence is assumed to be WIS with weights constant in each block. We shall do so as it simplifies the reasoning. Recall that the sorting Lemma 5 works for both right and left arguments.

Lemma 6. *Let $\boldsymbol{z}$ be an arbitrary vector in the domain of a Bregman divergence $D_\phi\big(\cdot \big|\big| \cdot\big)$ and $\boldsymbol{y}$ be partitioned as $\begin{bmatrix}\boldsymbol{y}_1 \\ \boldsymbol{y}_2\end{bmatrix}$. Let $\begin{bmatrix}\boldsymbol{z}_1 \\ \boldsymbol{z}_2\end{bmatrix}$ denote the conformal partition of $\boldsymbol{z}$. Let $D_\phi\big(\cdot \big|\big| \cdot\big)$ be a separable WIS Bregman divergence where the weights are constant within the partitions.*

[4] While one may address the discreteness problem via a real-relaxation of $\mathbb{P}$ to doubly stochastic matrices, the local minima attained in such a case will be in the interior of the Birkhoff polytope and not at a vertex of the polytope that is representable by $\mathbb{P}_i$ and reachable by sorting based updates (3.9). Therefore such a relaxation cannot answer whether (3.10) and sorting (3.9) achieves the local minimum of the cost function for a fixed $\{\boldsymbol{w}, \beta_i\}$ Surprisingly enough, sorting followed by a single $\boldsymbol{x}_i$ update achieves the local minimum of (3.4) on the cone $\mathfrak{R}_i$.

Let

$$\begin{bmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{bmatrix}^* = \mathrm{Argmin}_{\substack{\boldsymbol{y}'_i\in\Pi(\boldsymbol{y}_i),\\ \boldsymbol{y}'_1\geq\boldsymbol{y}'_2}} D_\phi\big(\begin{bmatrix}\boldsymbol{y}'_1\\ \boldsymbol{y}'_2\end{bmatrix}\Big|\Big|\begin{bmatrix}\boldsymbol{z}_1\\ \boldsymbol{z}_2\end{bmatrix}\big)$$

where $\Pi(\boldsymbol{y}_i)$ *is the set of all permutations of the vector* $\boldsymbol{y}_i$*, then* $\boldsymbol{y}_i^*$ *is isotonic with* $\boldsymbol{x}_i$ $\forall i = 1, 2$

Proof. The proof is by contradiction. Assume $\boldsymbol{y}_i^*$ is a minimizer that is not isotonic with $\boldsymbol{z}_i$, then according to lemma 5 one may permute $\boldsymbol{y}_i^*$ to match the order of $\boldsymbol{z}_i$ to reduce the divergence further, yielding a contradiction. □

Thus in spite of the caveats mentioned above, one can identify the optimal ordering of the components of the left argument that achieves the minimum for a fixed $\boldsymbol{w}^{t+1}, \{\beta_i^{t+1}\}$ even before optimal $\boldsymbol{x}_i$ has been determined. With this optimal order obtained, one may then compute the optimal $\boldsymbol{x}_i$ (see (3.11)) for a fixed $\boldsymbol{w}^{t+1}, \{\beta_i^{t+1}\}$ with relative ease using any convex optimization solver (in our experiments we use LBFGS (Liu et al., 1989)).

Block Equivalent Partial Order

Without any loss of generality we represent $\mathfrak{R}_i$ as the image of $\tilde{U}\boldsymbol{x} = M_i U_i \boldsymbol{x}$ where $\boldsymbol{x} \in {\mathbb{R}_+}^{k_i}$. U_i is an upper triangular matrix, similar in spirit to U in Lemma 3, but of size $k_i \times k_i$. For the constraint Δ_o^i we use matrix $\tilde{T}_i = M_i T_i$ instead of $\tilde{U}_i$ and constrain $\boldsymbol{x}$ to Δ. The run length decoding matrix ${M_i}^\dagger = \begin{bmatrix} 110\cdots0 \\ 0011\cdots \\ \cdots\cdots \end{bmatrix}$ is structured to select components of $U_i\boldsymbol{x}$ (or $T_i\boldsymbol{x}$) and copy them at the right position.

For the partially hidden order case (Section 3.3.1) the algorithm (Figure 3.1) can exploit the fact that multiplication by U_i (or T_i) or its inverse is a linear time operation. Therefore, a pertinent concern is whether something similar holds for the block equivalent partial order scenario for solving $\min_{\boldsymbol{x}} D_\phi\big(\tilde{U}_i\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big)$ and the the solution of the equation $\tilde{U}_i\boldsymbol{x} = \tilde{\boldsymbol{y}}$. The rank deficiency of $\tilde{U}$ seems troublesome. Indeed, the corresponding computations for the block equivalent partial order case too can be obtained efficiently thanks

to favorable properties of Bregman divergences, to whit: we present the following semi-closed form: It is easy to see that multiplication by $\tilde{U}_i$ is $O(d_i)$ because it consists of a k_i dimensional $\mathrm{Cum\text{-}Sum}$ and redistribution to obtain a vector in $\mathbb{R}^{d_i}$.

Lemma 7. *Given an IS Bregman divergence,*

$$\underset{\boldsymbol{x}\in{\mathbb{R}_+}^{k_i}}{\mathrm{Argmin}} D_\phi\big(\tilde{U}_i\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big) = \{\boldsymbol{x}^*|U_i\boldsymbol{x} = \mathrm{Proj}^\phi\left(\mu_\phi(\tilde{\boldsymbol{r}}_i), \mathcal{R}{\downarrow_{k_i}}\right)\}$$

and

$$\underset{\boldsymbol{x}\in\Delta}{\mathrm{Argmin}} D_\phi\big(\tilde{T}_i\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big) = \{\boldsymbol{x}^*|T_i\boldsymbol{x} = \mathrm{Proj}^\phi\left(\mu_\phi(\tilde{\boldsymbol{r}}_i), \Delta_{ok_i}\right)\}$$

Proof. Let $\mathbb{R}^{k_i} \ni \boldsymbol{q} = U_i\boldsymbol{x}$. The cost function reduces to

$$\begin{aligned}
&\min_{\boldsymbol{q}\in\mathcal{S}} \sum_{k=1}^{k_i}\sum_{j\in P_k} D_\phi\big(q_k\Big|\Big|y_j\big) \\
&\quad\overset{(a)}{=} \min_{\boldsymbol{q}\in\mathcal{S}} \sum_{k=1}^{k_i}\sum_{j\in P_k} D_\psi\big(\nabla\phi(y_j)\Big|\Big|\phi(q_k)\big) && (3.12)\\
&\quad\overset{(b)}{=} \min_{\boldsymbol{q}\in\mathcal{S}} \sum_{k=1}^{k_i}\sum_{j\in P_k} D_\psi\big(\phi(y_j)\Big|\Big|\mathop{\mathbb{E}}_{j\in P_k}[\nabla\phi(y_j)]\big) + \sum_{k=1}^{k_i} D_\psi\big(\mathop{\mathbb{E}}_{j\in P_k}[\nabla\phi(y_j)]\Big|\Big|\phi(q_k)\big) && (3.13)\\
&\quad\overset{(c)}{=} \sum_{k=1}^{k_i}\sum_{j\in P_k} D_\psi\big(\phi(y_j)\Big|\Big|\mu_\phi(\boldsymbol{y}_{P_k})\big) + \min_{\boldsymbol{q}\in\mathcal{S}} D_\phi\big(q_k\Big|\Big|\mu_\phi(\boldsymbol{y}_{P_k})\big) && (3.14)
\end{aligned}$$

Equality (a) follows from switching argument order identity (2.4), (b) from optimality of mean (A.7) and (c) from Corollary (8). The first term in (3.12) is constant hence the minimizer is obtained by minimizing the second term over the appropriate set $\mathcal{S}$ specified, obtaining the projection. □

This reduces the optimization problem into a Bregman projection problem of a significantly reduced dimensionality. The updates are shown in Figure 3.2. Back-solving with U_i is $O(k_i)$ and computing $\mu_\phi(\tilde{\boldsymbol{r}}_i)$ is $O(d_i)$ if $\nabla\phi$ and $\nabla^{-1}\phi$ can be computed in constant

Input: Convex function ϕ, feature matrices $\{A_i\}$ with rows sorted by relevance, regularization parameter C.

Repeat Until Convergence:

$$\boldsymbol{x}_i^{t+1} = \{\boldsymbol{x}^* | T_i \boldsymbol{x} = \text{Proj}^{\phi}\left(\mu_{\phi}(\tilde{\boldsymbol{r}}_i), \Delta_{ok_i}\right)\} \tag{3.15}$$

$$\boldsymbol{w}^{t+1}, \{\beta_i^{t+1}\} = \underset{\boldsymbol{w}}{\text{Argmin}} \sum_{i=1}^{|\mathcal{Q}|} D_{\boldsymbol{\phi}}\big(\tilde{T}_i \boldsymbol{x}^{t_i+1} \Big|\Big| (\nabla\phi)^{-1}\left(\boldsymbol{A}_i{}^{\dagger}\boldsymbol{w} + \beta_i^t \mathbf{1}\right)\big) \tag{3.16}$$

Return: $\boldsymbol{w}$.

Figure 3.2: Algorithm for Block Equivalent Partial Order

time. If $\boldsymbol{\phi}$ belongs to the "essentially smooth" class, e.g. wKL, wGI, Corollary 1 can reduce computation even further.

3.4 Experiments

We evaluated the ranking performance of the proposed monotone retargeting approach on the benchmark LETOR 4.0 datasets (MQ2007, MQ2008) (Liu et al., 2007) as well as the OHSUMED dataset (Hersh et al., 1994). Each of these datasets is pre-partitioned into five-fold validation sets for easy comparison across algorithms. For OHSUMED, we used the *QueryLevelNorm* partition. Each dataset contains a set of queries, where each document is assigned a relevance score from irrelevant ($r = 0$) to relevant ($r = 2$).

All algorithms were trained using a regularized linear regression function, with a regularization parameter chosen from the set $C \in \{10^{-20}, 10^{-10}, 10^{-5}, 10^{0}, 10^{1}\}$. The best model was identified as the model with highest mean average precision (MAP) on the validation set. All presented results are of average performance on the test set. As the baseline, we implemented the NDCG consistent re-normalization approach in (Ravikumar et al., 2011) (using the NDCG$_m$ normalization) for the squared loss and the I-divergence (generalized KL-divergence). The baseline constitutes the latest state of the art in super-

vised ranking methods. It incorporates NDCG consistency into the formulation and was recently shown to outperform the then state of the art LETOR algorithms Listnet (Cao et al., 2007a), RankCosine (Ravikumar et al., 2011) and other NDCG inconsistent metrics, see (Ravikumar et al., 2011) for details.

ListNet was implemented (Cao et al., 2007a) as the KL divergence baseline since their normalization has no effect on KL-divergence. MR was implemented using the *partially hidden order* monotone retargeting approach (Section 3.3.1). We compared the performance of MR (Normalized MR) to the MR method with the normalization $\frac{1}{|\mathcal{V}_i|}$ removed (Unnormalized MR).

The algorithms were implemented in Python and executed on a 2.4GHz quad-core Intel Xeon processor without paying particular attention to writing optimized code. Ample room for improvement remains. Square loss was the fastest with respect to average execution times per iteration at 0.58 seconds whereas KL achieved 1.01 seconds per iteration and I-div 1.14 seconds per iteration. We found that although MQ2007 is more than 4 times larger than MQ2008, MQ2007 only required about twice the time execution on average, highlighting the scalability of MR. On average SQ, KL and I-div took 99, 90 and 65 iterations.

Table 3.4 compares the algorithms in terms of expected reciprocal return (ERR) (Chapelle et al., 2009), mean average precision (MAP) and NDCG. The unnormalized KL divergence cost function led to the best performance across datasets. The most significant gains over the baseline were for the I-divergence cost function. Monotone retargeting showed consistent performance gains over the baseline across metrics (NDCG, ERR, Precision), suggesting the effectiveness of MR for improving the overall ranking performance.

Figures 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 show the performance characteristics measured according to NDCG@N and Precision@N metrics of MR with I-divergence, KL-divergence and Sq-loss and the corresponding state of the art baselines. Our experiments

[4] The baselines are obtained by applying NDCG consistency correction of Ravikumar et al. (2011) to the base models and were shown to outperform then state of the art algorithms such as ListNet,RankCosine etc.

MQ 2007 NDCG			
	I-div	SQ	KL
Unnormalized MR	0.6961	**0.7398**	0.6978
Normalized MR	0.6954	0.6953	0.6981
Baseline[4]	0.5512	0.6927	0.6952
MQ 2007 MAP			
	I-div	SQ	KL
Unnormalized MR	0.5379	0.5361	0.5398
Normalized MR	0.5358	0.5282	**0.5399**
Baseline[4]	0.3611	0.5330	0.5380
MQ 2007 ERR			
	I-div	SQ	KL
Unnormalized MR	0.3698	0.3703	**0.3737**
Normalized MR	0.3702	0.3601	0.3731
Baseline[4]	0.1953	0.3639	0.3643

Table 3.2: Test NDCG, MAP and ERR on dataset MQ 2007. The best results are noted in bold.

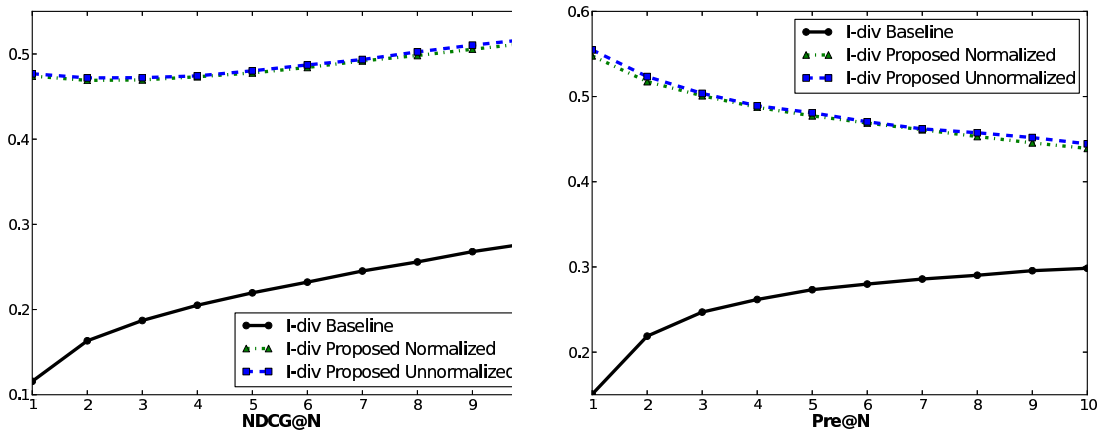


Figure 3.3: NDCG (left) and Precision (right) on MQ2007 obtained by MR with I-divergence and I-divergence based baselines.

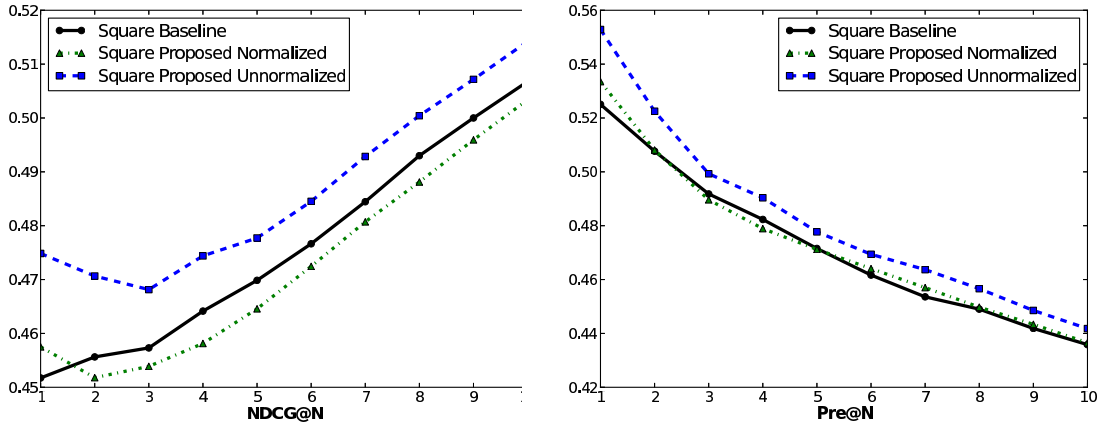


Figure 3.4: NDCG (left) and Precision (right) MQ2007 obtained by MR with sq-loss and sq-loss based baselines.

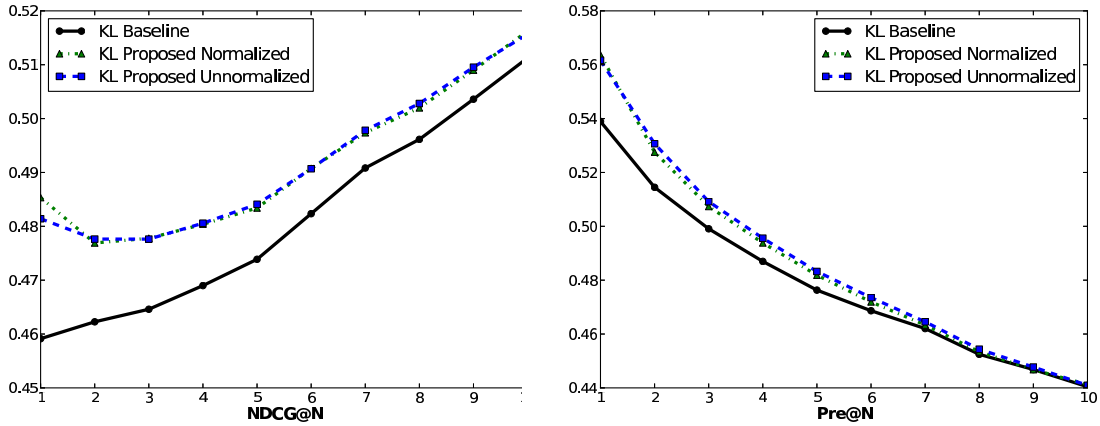


Figure 3.5: NDCG (left) and Precision (right) on MQ2007 obtained by MR with KL-divergence and KL-divergence based baselines.

MQ 2008 NDCG			
	I-div	SQ	KL
Unnormalized MR	0.7339	0.7398	**0.7451**
Normalized MR	0.7346	0.7396	0.7330
Baseline[4]	0.5892	0.7344	0.7399
MQ 2008 MAP			
	I-div	SQ	KL
Unnormalized MR	0.6439	0.6532	**0.6571**
Normalized MR	0.6449	0.6549	0.6461
Baseline[4]	0.4513	0.6428	0.6530
MQ 2008 ERR			
	I-div	SQ	KL
Unnormalized MR	0.4137	0.41559	**0.4238**
Normalized MR	0.4144	0.41392	0.4085
Baseline[4]	0.2724	0.40978	0.4132

Table 3.3: Test ERR, MAP and NDCG on MQ2008 dataset. The best results are noted in bold.

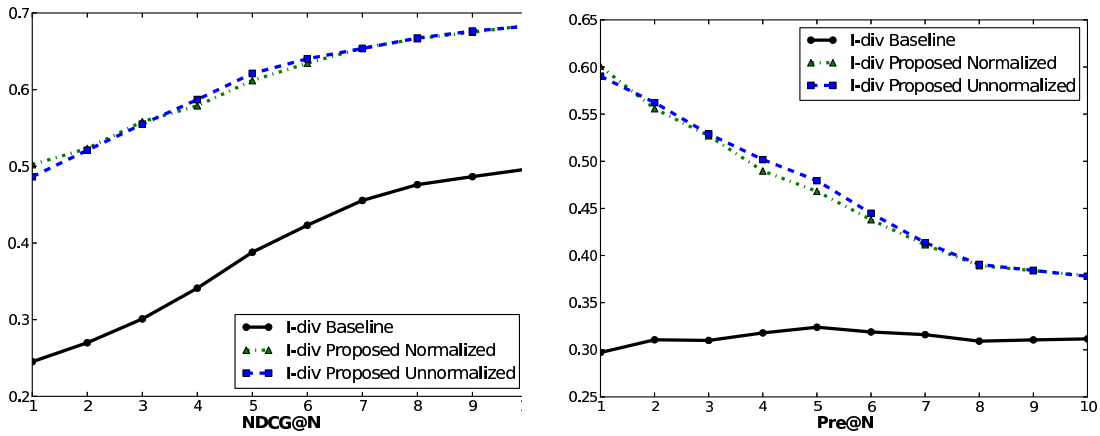


Figure 3.6: NDCG (left) and Precision (right) on MQ2008 obtained by MR with I-divergence and Idivergence based baselines.

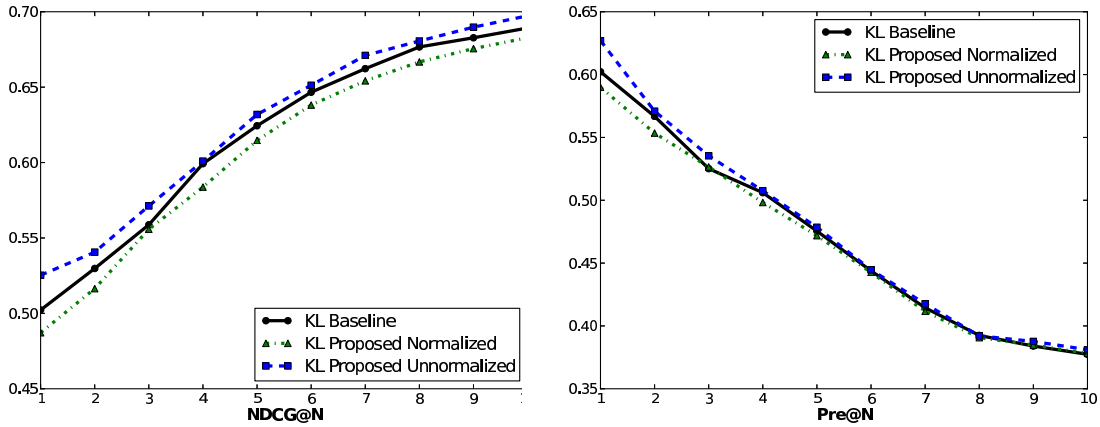


Figure 3.7: NDCG (left) and Precision (right) on MQ2008 obtained by MR with KL-divergence and KL-divergence based baselines.

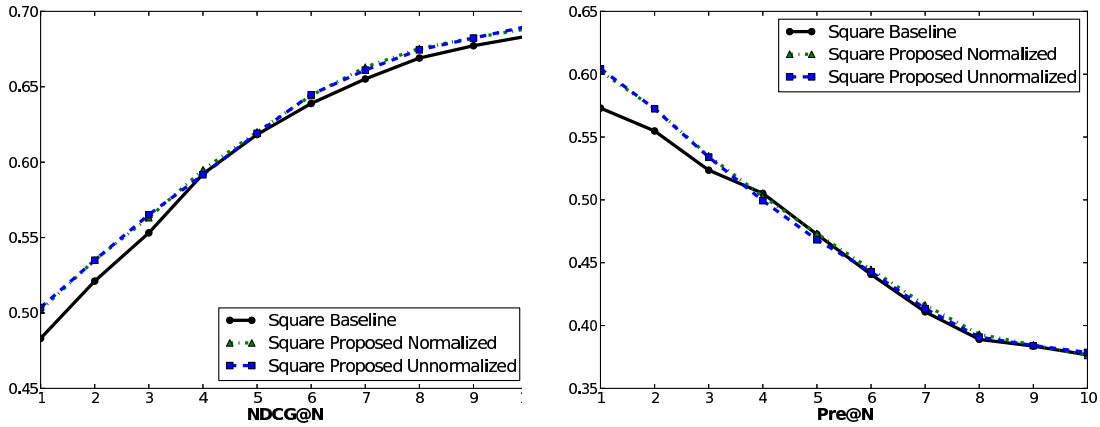


Figure 3.8: NDCG and Precision on MQ2008 obtained by MR with sq-loss and sq-loss based baselines.

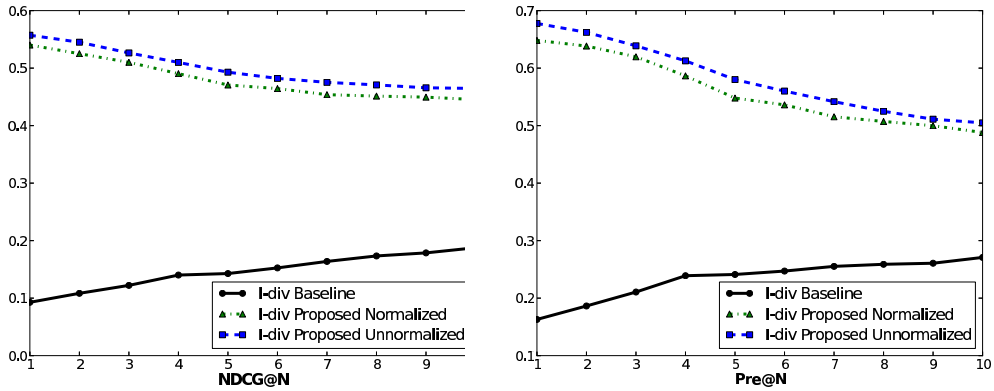


Figure 3.9: NDCG (left) and Precision (right) on OHSUMED obtained by MR with I-divergence and I-divergence based baselines.

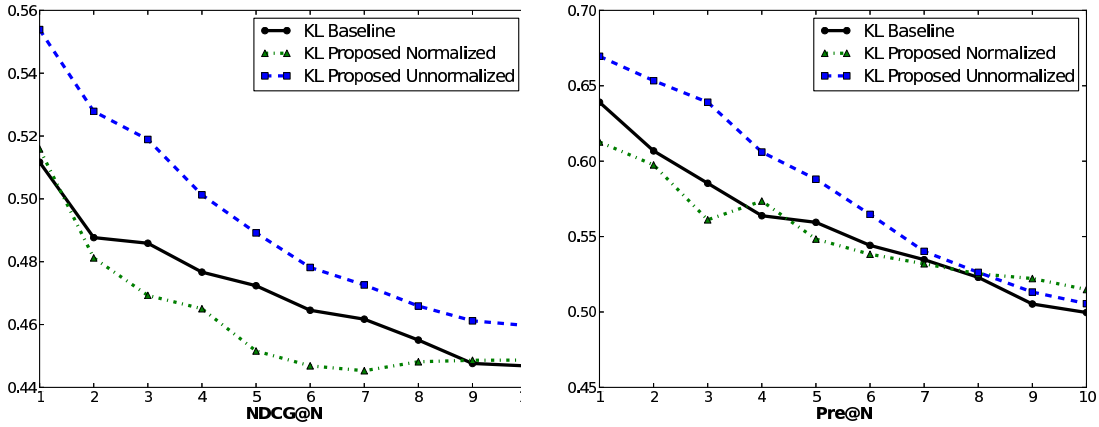


Figure 3.10: NDCG (left) and Precision (right) on OHSUMED obtained by MR with I-divergence and I-divergence based baselines.

OHSUMED ERR			
	I-div	SQ	KL
Unnormalized MR	0.5657	0.5410	0.5410
Normalized MR	**0.5796**	0.5093	0.5093
Baseline[4]	0.2255	0.5450	0.5467
OHSUMED MAP			
	I-div	SQ	KL
Unnormalized MR	**0.4537**	0.4417	0.4531
Normalized MR	0.4463	0.4394	0.4506
Baseline[4]	0.3421	0.4465	0.4524
OHSUMED NDCG			
	I-div	SQ	KL
Unnormalized MR	**0.7000**	0.6878	0.6997
Normalized MR	0.6935	0.6798	0.6916
Baseline[4]	0.5805	0.6892	0.6947

Table 3.4: Test ERR, MAP and NDCG on OHSUMED dataset. The best results are in bold.

show a significant improvement in performance on the range of datasets and cost functions. Across datasets, the difference between the baseline and our results were most significant with the I-divergence (generalized KL divergence) cost function.

There are two things worth taking special note of: (i) although the baseline algorithms were proposed specifically for improving NDCG performance, MR improves the ranking accuracy further, even in terms of NDCG. (ii) MR seems to be achieving peak performance early, consistently. This property is particularly desirable and is encoded specifically in the cost functions such as NDCG and ERR. In our initial formulation we used WIS Bregman divergence so that the weights could be tuned to obtained the early peaking behavior. However that proved unnecessary because even the unweighted model produced satisfactory performance. The effect of query length normalization was, however, inconsistent. Some of our results were insensitive to it, whereas other results were adversely affected. We conjecture that the restriction of the scores to the unit simplex already normalizes the qsets based on item sizes and thus additional normalization is unnecessary.

3.4.1 Joint Convexity

Now we extend the property of joint convexity beyond squared Euclidean distance. This can be done using a careful balance between regularizing $\boldsymbol{r}_i$ and $\boldsymbol{w}$. We regularize $\boldsymbol{r}_i$ via the term $C_{ri} D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla\phi)^{-1}(\boldsymbol{q}_i)\big)$ to ensure joint convexity. Necessary and sufficient conditions for are established for the coefficient C_{ri}.

Vector $(\nabla\boldsymbol{\phi})^{-1}(\boldsymbol{q}_i)$ acts as the "center" of regularization for $\boldsymbol{r}_i$. We use $\tilde{\boldsymbol{r}}_i = (\nabla\boldsymbol{\phi})^{-1}(\boldsymbol{q}_i)$ in the batch setting and $\operatorname{Argmin}_{\boldsymbol{x}} \boldsymbol{\phi}$ in the online setting. Incorporating this regularization we obtain

$$F_i(\boldsymbol{r}_i, \boldsymbol{w}) = \frac{1}{|\mathcal{V}_i|}\Big(D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla\phi)^{-1}(\boldsymbol{A}_i\boldsymbol{w})\big) + C_{ri} D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla\phi)^{-1}(\boldsymbol{q}_i)\big) + \frac{C_{wi}}{2}||\boldsymbol{w}||^2_{\boldsymbol{A}_i} + \frac{C|\mathcal{V}_i|}{2|\mathcal{Q}|}||\boldsymbol{w}||^2\Big) \quad (3.17)$$

Our first update of the cost function (3.4) is $F(\{\boldsymbol{r}_i\}, \boldsymbol{w}) = \sum_i^{|\mathcal{Q}|} F_i(\boldsymbol{r}_i, \boldsymbol{w})$. Note that β terms may be absorbed into $\boldsymbol{A}_i$ by augmenting the features by a vector of ones, so no generality is lost in equation (3.17) and that we assume ϕ to be strongly convex.

Lemma 8. *Let ϕ be s strongly convex with L Lipschitz continuous gradients, then maintaining $C_{wi} + \frac{1}{L} > 0$,*

$s(C_{ri}+1)(C_{wi}+\frac{1}{L}) \geq 1$ *ensures F_i is jointly convex.*

Proof. $\nabla^2 F_i(\boldsymbol{r}_i, \boldsymbol{w}) = \frac{1}{|\mathcal{V}_i|}\begin{bmatrix} (1+C_{ri})H_\phi & -\boldsymbol{A} \\ -\boldsymbol{A}^\dagger & {\boldsymbol{A}_i}^\dagger (H_\psi + C_{wi})\boldsymbol{A}_i + \frac{C|\mathcal{V}_i|}{2|\mathcal{Q}|} I \end{bmatrix}$, where ψ is the Legendre conjugate of ϕ and H_ϕ, H_ψ the corresponding diagonal [5] Hessians. Substitutin g the relation between C_{wi}, C_{ri} and bounding the smallest eigenvalue, the result follows. □

Lemma 9. *Let $\alpha_i = \frac{1}{1+C_{\phi_i}}$, then*

$$\operatorname*{Argmin}_{\boldsymbol{r}_i \in \mathcal{R}\downarrow_i \cap \mathcal{S}_i} F_i(\boldsymbol{r}_i, \boldsymbol{w}) = \operatorname*{Argmin}_{\boldsymbol{r}_i \in \mathcal{R}\downarrow_i \cap \mathcal{S}_i} D_{\boldsymbol{\phi}}\big(\boldsymbol{r}_i \big|\big| (\nabla\phi)^{-1}(\alpha_i \boldsymbol{A}_i \boldsymbol{w} + (1-\alpha_i)\boldsymbol{q}_i)\big).$$

[5] Recall that $\boldsymbol{\phi}(\cdot)$ and consequently $\boldsymbol{\psi}(\cdot)$ are separable by assumption.

Proof. Follows as a consequence of $\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\boldsymbol{\pi}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{s}\big)\right] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\boldsymbol{\pi}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{\mu}\big)\right] + D_\phi\big(\boldsymbol{\mu}\big\|\boldsymbol{s}\big)$ (Banerjee et al., 2005). □

Thus $\boldsymbol{r}_i$'s can be updated using *deflected* prediction $(\nabla\phi)^{-1}(\alpha A\boldsymbol{w} + (1-\alpha)\boldsymbol{q}_i)$. Strong convexity of ϕ is a mild assumption that is satisfied by all the com monly used Bregman divergences, e.g. squared Euclidean, KL-divergence, I-divergence etc.

3.4.2 Marginal Strong Convexity

Since $F(\{\boldsymbol{r}_i\}, \boldsymbol{w})$ is jointly convex we may work with the marginal function

$$G_i(\boldsymbol{w}) = \min_{\{\boldsymbol{r}_i\}} F(\{\boldsymbol{r}_i\}, \boldsymbol{w}), \quad G(\boldsymbol{w}) = \sum_i G_i(\boldsymbol{w}) \tag{3.18}$$

which is guaranteed to be convex (Rockafellar, 1996). This luxury is not availabl e in MR. A quasi-Newton method (Liu et al., 1989) applied to $G(\boldsymbol{w})$ would require computing $\nabla G(\boldsymbol{w})$, this is easily obtained as

$$\nabla G(\boldsymbol{w}) = \sum_i^{|\mathcal{Q}|} G_i(\boldsymbol{w}) = \sum_i^{|\mathcal{Q}|} \nabla F_i(\{\boldsymbol{r}_i^*\}, \boldsymbol{w}) \tag{3.19}$$

where $\boldsymbol{r}_i^* = \mathrm{Argmin}_{\boldsymbol{r}_i\in\mathcal{R}_\downarrow} F_i(\boldsymbol{r}_i, \boldsymbol{w})$. Observe that the *gradient computation trivially parallelizes because the $\boldsymbol{r}_i$s are all independent*. It is indeed beneficial for $G(\boldsymbol{w})$ to be convex, but strong convexity of $G_i(\boldsymbol{w})$ would further facilitate super-linear convergence of quasi-Newton methods, and guarantee logarithmic regret in the online setting (Hazan et al., 2007). Using assumptions of continuous second order differentiability and the shorthand $F_i^* = F_i(\boldsymbol{r}_i^*, \boldsymbol{w})$ we obtain

$$\nabla^2 G_i(\boldsymbol{w}) = \nabla^2_{\boldsymbol{w}} F_i^* - \nabla_{\boldsymbol{w},\boldsymbol{r}_i} F_i^{*\dagger} (\nabla^2_{\boldsymbol{r}_i} F_i^*)^{-1} \nabla_{\boldsymbol{w},\boldsymbol{r}_i} F_i^* = \frac{\boldsymbol{A}_i^\dagger}{|\mathcal{V}_i|}\left[H_\psi + C_{wi} - \frac{1}{1+C_{\phi_i}}(H_\phi)^{-1}\right]\boldsymbol{A}_i + \frac{C}{|\mathcal{Q}|} I \tag{3.20}$$

Lemma 10. *Conditions of Lemma 8 ensure that $G(\boldsymbol{w})$ is C strongly convex and $\nabla G(\boldsymbol{w})$ is $\sum_i \frac{\sigma_i}{|\mathcal{V}_i|}(C_{wi} - \frac{1}{s(1+C_{ri})}) + C$ Lipschitz continuous, where σ_i is the singular value of $\boldsymbol{A}_i$.*

3.4.3 Lipschitz Continuity of Hessian

In order to enjoy local quadratic convergence, quasi-Newton methods require that the objective function (i) is twice differentiable, (ii) is strongly convex and (iii) has Lipschitz continuous Hessians (Boyd and Vandenberghe, 2004). For $G(\boldsymbol{w})$ the first two criteria holds directly, here we explore when is the third satisfied. Observe from equation (3.20) that we only need to be concerned about t he sensitivity of the term $\left[H_\psi + C_{wi} - \frac{1}{1+C_{\phi_i}}(H_\phi)^{-1}\right]$ to va riations in $\boldsymbol{w}$. We make the notation more precise about dependency on $\boldsymbol{w}$. Let $\boldsymbol{r}_i^*(\boldsymbol{w}) = \mathrm{Argmin}_{\boldsymbol{r}_i \in \mathcal{R}\downarrow} F_i(\boldsymbol{r}_i, \boldsymbol{w})$ and the paranthesis indicate where the Hessians are evaluated in: $\left[H_\psi(\boldsymbol{w}) + C_{wi} - \frac{1}{1+C_{\phi_i}}(H_\phi(\boldsymbol{r}_i^*(\boldsymbol{w})))^{-1}\right]$.

Lemma 11. *Let $\psi(\cdot)$ be the Legendre conjugate of $\phi(\cdot)$ that defines the cost function $G(\boldsymbol{w})$ in equation* (3.18)*. Then if $\psi(\cdot)$ has a Lipschitz continuous Hessian then $G(\boldsymbol{w})$ has a Lipschitz continuous Hessian.*

Proof. $\left[H_\psi(\boldsymbol{w})+C_{wi}-\frac{1}{1+C_{\phi_i}}(H_\phi(\boldsymbol{r}_i^*(\boldsymbol{w})))^{-1}\right] = \left[H_\psi(\boldsymbol{w})+C_{wi}-\frac{1}{1+C_{\phi_i}}H_\psi(\nabla\phi(\boldsymbol{r}_i^*(\boldsymbol{w})))\right]$ using Legendre duality. Further, the vector $\nabla\phi(\boldsymbol{r}_i^*(\boldsymbol{w}))$ turns out to be the Euclidean projection of the vector $\boldsymbol{A}_i\boldsymbol{w}$ on the set $\mathcal{R}\downarrow_i$ (see Proposition2). Now, since projection is a non-expansive operator, $H_\psi(\nabla\phi(\boldsymbol{r}_i^*(\boldsymbol{w})))$ is Lipschitz continuous in variations in $\boldsymbol{w}$. □

3.4.4 Margins on Target Vectors

We now augment the cost function by introducing a pair of fixed margin (3.21), (3.22) and a pair of large margin variants (3.23), (3.24). We enforce an order in the target vector $\boldsymbol{r}_i$ but also enforce a gap between the target values of two adjacently ordered items $r_{i,j}$, $r_{i,j+1}$.

Since our modification takes the form of addition of linear inequalities and terms, the properties of strong convexity and Lipschitz continuity of the gradient continue to hold. By controlling the margin we can model the notion that errors at the top of the list are more severe than at the bottom. More separated the targets, higher the tendency of the regression function to maintain the separation and, consequently, the order.

The **fixed margin formulations** are posed in terms of positive pre-prescribed margins $t_{i,j}$ as follows:

$$\min_{\boldsymbol{r}_i,\boldsymbol{w}} \sum_{i=1}^{|\mathcal{Q}|} F_i(\boldsymbol{r}_i,\boldsymbol{w}) \;\; \text{s.t.} \;\; \{r_{i,j+1} - r_{i,j} \geq t_{i,j}\}_{\substack{\forall j \in [0,d_i-1]; \\ \forall i \in [1,|\mathcal{Q}|]}} \\ \{r_{i,0} \geq t_{i,0}\}_{\forall i \in [1,|\mathcal{Q}|]} \tag{3.21}$$

$$\min_{\boldsymbol{r}_i,\boldsymbol{w}} \sum_{i=1}^{|\mathcal{Q}|} F_i(\boldsymbol{r}_i,\boldsymbol{w}) \;\; \text{s.t.} \;\; \{r_{i,j+1} - r_{i,j} \geq t_{i,j}\}_{\substack{\forall j \in [0,d_i-1]; \\ \forall i \in [1,|\mathcal{Q}|]}} \\ \{r_{i,d_i} \leq t_{i,d_i}\}_{\forall i \in [1,|\mathcal{Q}|]}. \tag{3.22}$$

The **large margin formulations** are posed in terms of a vector of *rewards* $\boldsymbol{c}_i$ associated with the vector of gaps $\boldsymbol{t}_i > \boldsymbol{0}$ as follows: for every query $q_i i \in \mathcal{Q}$, solve:

$$\min_{\boldsymbol{r}_i,\boldsymbol{w},\boldsymbol{t}_i} \sum_{i=1}^{|\mathcal{Q}|} F_i(\boldsymbol{r}_i,\boldsymbol{w}) - \langle \boldsymbol{c}_i, \boldsymbol{t}_i \rangle \;\; \text{s.t.} \;\; \{r_{i,j+1} - r_{i,j} \geq t_{i,j} \geq 0\}_{\substack{\forall j \in [0,d_i-1]; \\ \forall i \in [1,|\mathcal{Q}|]}} \\ \{r_{i,0} \geq t_{i,0}\}_{\forall i \in [1,|\mathcal{Q}|]}, \tag{3.23}$$

$$\min_{\boldsymbol{r}_i,\boldsymbol{w},\boldsymbol{t}_i} \sum_{i=1}^{|\mathcal{Q}|} F_i(\boldsymbol{r}_i,\boldsymbol{w}) - \langle \boldsymbol{c}_i, \boldsymbol{t}_i \rangle \;\; \text{s.t.} \;\; \{r_{i,j+1} - r_{i,j} \geq t_{i,j} \geq 0\}_{\substack{\forall j \in [0,d_i-1]; \\ \forall i \in [1,|\mathcal{Q}|]}} \\ \{r_{i,d_i} \leq t_{i,d_i}\}_{\forall i \in [1,|\mathcal{Q}|]}. \tag{3.24}$$

In all the formulations (3.21), (3.22), (3.23), (3.24) the components of $\boldsymbol{t}_i$ denote the gap between the adjacent targets. In (3.21) and (3.22) the gaps are pre-specified. It is natural to specify a comparatively higher gap at the top. In (3.24) and (3.23) the gaps are not specified explicitly, but a reward $\boldsymbol{c}_i$ is awarded per unit gap.

The optimization over $\boldsymbol{w}$ is regularized maximum likelihood parameter estimation

for GLMs (McCulloch and Searle, 2001). Since this procedure is standard, we will focus on $\boldsymbol{r}$ and $\boldsymbol{t}$.

3.4.5 Bregman Projection on $\mathcal{R}\downarrow_t$

If we fix $\boldsymbol{t}$ and $\boldsymbol{w}$ in equations (3.21), (3.22), (3.23), (3.24) we obtain the following problem on $\boldsymbol{r}$:

$$\min_{\boldsymbol{r}} D_\phi\big(\boldsymbol{r}\Big|\Big|(\nabla\phi)^{-1}(A\boldsymbol{w})\big) \text{ s.t. } \text{Adj-Diff}\,(\boldsymbol{r}) \leq \boldsymbol{t}. \tag{3.25}$$

Can (3.25) be reduced to a squared loss minimization problem ? Under assumptions of strong convexity and/or Lipschitz continuity of ϕ we can respond in the affirmative.

Proposition 2. *Let $\phi(\cdot)$ be s strongly convex, then*

$$(\nabla\phi)^{-1}(\boldsymbol{z}^*) = \text{Argmin}_{\boldsymbol{r}}\, D_\phi\big(\boldsymbol{r}\Big|\Big|(\nabla\phi)^{-1}(A\boldsymbol{w})\big) + \langle \boldsymbol{v}, \boldsymbol{r}\rangle \text{ s.t. Adj-Diff}\,(\boldsymbol{r}) \leq \boldsymbol{t} \tag{3.26}$$

$$\textit{where} \quad \boldsymbol{z}^* = \text{Argmin}_{\boldsymbol{z}}\, ||\boldsymbol{z} - A\boldsymbol{w}|| + \langle \boldsymbol{v}, \boldsymbol{r}\rangle \quad \textit{s.t.} \quad \text{Adj-Diff}(\boldsymbol{z}) \leq s\boldsymbol{t}. \tag{3.27}$$

Proof. For the moment let us ignore the linear term $\langle \boldsymbol{v}, \boldsymbol{r}\rangle$. Let the set of points satisfying the KKT conditions for (3.25) be

$$\mathcal{A} = \left\{ \begin{matrix} \boldsymbol{r} \\ \boldsymbol{\lambda} \end{matrix} \middle| \begin{matrix} \nabla\phi(\boldsymbol{r}) = A\boldsymbol{w} - \text{Adj-Diff}(\boldsymbol{\lambda}), \\ \text{Adj-Diff}(\boldsymbol{r}) \leq \boldsymbol{t} \end{matrix} \right\}$$

and the set of points satisfying the KKT for $\min_{\boldsymbol{z}} ||\boldsymbol{z} - A\boldsymbol{w}||$ s.t. $\text{Adj-Diff}(\boldsymbol{z}) \leq c\boldsymbol{t}$ be

$$\mathcal{B} = \left\{ \begin{matrix} \boldsymbol{z} \\ \boldsymbol{\lambda} \end{matrix} \middle| \begin{matrix} \boldsymbol{z} = A\boldsymbol{w} - \text{Adj-Diff}(\boldsymbol{\lambda}) \\ \text{Adj-Diff}(\boldsymbol{z}) \leq c\boldsymbol{t} \end{matrix} \right\} = \left\{ \begin{matrix} \nabla\phi(\boldsymbol{r}) \\ \boldsymbol{\lambda} \end{matrix} \middle| \begin{matrix} \nabla\phi(\boldsymbol{r}) = A\boldsymbol{w} - \text{Adj-Diff}(\boldsymbol{\lambda}) \\ \text{Adj-Diff}(\nabla\phi(\boldsymbol{r})) \leq c\boldsymbol{t} \end{matrix} \right\}$$

(the latter is obtained by simple change of variables). From $r_{j+1} - r_j \geq t_j$ and strong convexity we have $\nabla\phi(r_{j+1}) - \nabla\phi(r_j) \geq st_j$ thus $\mathcal{A} \subset \mathcal{B}$. $\mathcal{A}, \mathcal{B}$ are unique minimizers, therefore the minima of the two problems coincide. The term $\langle \boldsymbol{v}, \boldsymbol{r}\rangle$ maintains the relation between $\mathcal{A}$ and $\mathcal{B}$. □

Proposition 3. *Let $\phi(\cdot)$ be strictly convex and let gradient $\nabla\phi(\cdot)$ be $\frac{1}{L}$ Lipschitz continuous, then minimi zer $\boldsymbol{z}^*$ of* (3.26) *is* $\boldsymbol{z}^* = \operatorname{Argmin}_{\boldsymbol{z}} ||\boldsymbol{z} - A\boldsymbol{w}|| + \langle \boldsymbol{v}, \boldsymbol{r}\rangle$ *s.t.* $\text{Adj-Diff}(\boldsymbol{z}) \leq L\boldsymbol{t}$.

Proof. Define $\mathcal{A}$ and $\mathcal{B}$ as before. From $\nabla\phi(r_{j+1}) - \nabla\phi(r_j) \geq Lt_j$ and Lipschitz continuity we have $r_{j+1} - r_j \geq t_j$ therefore $\mathcal{B} \subset \mathcal{A}$, but $\mathcal{A}$ and $\mathcal{B}$ are unique minimizers. □

It is critical to solve quadratic program (QP) in equations (3.27) efficiently because $\boldsymbol{r}$ minimization forms a part of the gradient computation (3.19) thus we cannot afford the expense of a generic QP solver in an inner loop. If $\boldsymbol{t} = \boldsymbol{0}$ the equivalent QP can remarkably be solved in linear time by the PAV (Grotzinger and Witzgall, 1984) algorithm. Its efficiency heavily depends on the blockwise constant structure of the optimal (Acharyya et al., 2012). No such structure is guaranteed for the QPs obtained by Proposition 2 and 3. Nevertheless, these too can be solved in linear time.

A key tool that we employ to obtain the solution efficiently is the pool adjacent violators algorithm, it solves

$$\min_{\boldsymbol{z}} ||\boldsymbol{z} - A\boldsymbol{w}|| \quad \text{s.t.} \quad \text{Adj-Diff}_*(\boldsymbol{z}) \leq \boldsymbol{0} \tag{3.28}$$

called the isotonic regression. PAV is essentially a block coordinate ascent of the dual of (3.28). It runs in *finite time*

Our interest lies in solving (3.21), (3.22), (3.23) and (3.24) which look drastically different from (3.28). We show that by a series of non-linear and linear change of variables one can reduce these problems to minor variations of the isotonic regression problem.

Decomposing the Max Margin Formulation

For a fixed $\boldsymbol{w}$, a plausible way to optimize (3.24) and (3.23) is to fix $\boldsymbol{t}_i$ and optimize $\boldsymbol{r}_i$ and alternate, keeping $\boldsymbol{w}$ fixed. One may update $\boldsymbol{w}$ once $\boldsymbol{t}_i$ and $\boldsymbol{r}_i$ converge. This fails to obtain the optimum because the constraints couple $\boldsymbol{r}_i$ and $\boldsymbol{t}_i$. However, we show that an affine transformation can not only correctly decompose the problem, but also separate

out the problem into versions of isotonic regression problems: namely isotonic regression with a lower-bound on the smallest r for (3.24) and isotonic regression with an upper-bound on the largest r for (3.23). Thus they add another (scalar) constraint to the system $\text{Adj-Diff}(\boldsymbol{r}) \leq -\boldsymbol{t}$. For convenience we denote both by $\text{Adj-Diff}^*(\boldsymbol{r}) \leq -\boldsymbol{t}$ to give them an unified treatment. Both the variants are solved in finite time by variations of the PAV algorithm (denoted by PAV^*) (Grotzinger and Witzgall, 1984) and the time scales linearly in dimension.

Because of Propositions 2, 3, we only need to consider:

$$\min_{\boldsymbol{r},\boldsymbol{t}} \frac{1}{2}||\boldsymbol{r}-\boldsymbol{y}||^2 - \langle \boldsymbol{c}, \boldsymbol{t} \rangle \text{ s.t. } \text{Adj-Diff}^*(\boldsymbol{r}) \leq -\boldsymbol{t}, \;\; \boldsymbol{t} > 0$$

for the maximum margin formulations. Substituting $\boldsymbol{t} = -\text{Adj-Diff}^*(\boldsymbol{d})$, $\boldsymbol{z} = \boldsymbol{r} - \boldsymbol{d}$ obtains

$$\frac{1}{2}||\boldsymbol{z}+\boldsymbol{d}-\boldsymbol{y}||^2+\langle \boldsymbol{c}, \text{Adj-Diff}^*(\boldsymbol{d}) \rangle \text{ s.t. } \quad \text{Adj-Diff}^*(\boldsymbol{z}) \leq 0, \quad \text{Adj-Diff}^*(\boldsymbol{d}) \leq 0. \tag{3.29}$$

The variables $\boldsymbol{z}$ and $\boldsymbol{d}$ are completely decoupled, the constraints are the ordering constraints, and if either $\boldsymbol{z}$ or $\boldsymbol{d}$ fixed, the other is a PAV problem. For $\boldsymbol{d}$, some algebraic manipulation is necessary to expose the PAV form. Thus, one may alternate over $\boldsymbol{z}$ and $\boldsymbol{d}$ as follows:

$$\boldsymbol{z}^{t+1} = PAV^*(\boldsymbol{y} - \boldsymbol{d}^t) \tag{3.30}$$

$$\boldsymbol{d}^{t+1} = PAV^*(\boldsymbol{y} - \boldsymbol{z}^{t+1} - \text{Adj-Diff}^{*\dagger}(\boldsymbol{c})) \tag{3.31}$$

and obtain the large margin solution by recovering $\boldsymbol{r}, \boldsymbol{t}$ from converged $\boldsymbol{z}$ and $\boldsymbol{d}$.

Decomposing the Fixed Margin Formulation

Problems (3.21), (3.22) can be decomposed similarly using Propositions 2, 3 and the exact same affine transformation $\boldsymbol{t} = -\text{Adj-Diff}^*(\boldsymbol{d})$ and $\boldsymbol{z} = \boldsymbol{r} - \boldsymbol{d}$. Here d is immediately

determined by equation $\boldsymbol{t} = -\text{Adj-Diff}^*(\boldsymbol{d})$, so no iteration over $\boldsymbol{z}$ and $\boldsymbol{d}$ is necessary . Solving $\boldsymbol{z} = PAV^*(\boldsymbol{y}-\boldsymbol{d})$ is sufficient to recover the optimal $\boldsymbol{r}$. Since this requires a single instance of PAV, it is obvious that this converges in finite time, linear in the dimension.

3.4.6 Convergence Rates for Batch and Online settings

Convergence rate guarantees are readily available for i) batch gradient descent with (3.19) evaluated in parallel. As a result of strong marginal convexity this converges linearly (Bertsekas, 1999). ii) Stochastic gradient descent by sampling an index from (3.19). Again strong convexity ensures that this has linear rate of convergence (in an expected sense) (Rakhlin et al., 2012). iii) Quasi-Newton and Newton methods with parallel evaluation of gradients: The former will only use the gradient computation (3.19), whereas the latter will use the explicit Hessian (3.20) which has a simple diagonal structure, with identity on the off diagonal blocks. These will have superlinear convergence (Bertsekas, 1999). In our experiments we use LBFGS (Liu et al., 1989) as our Quasi-Newton method. (iv) Finally, like in the MR paper (Acharyya et al., 2012) one can use block coordinate descent, that due to lemma 8 is guaranteed linear rate of convergence (Bertsekas, 1999). Here the $\boldsymbol{r}_i$ can be trivially parallelized because they are independent, for $\boldsymbol{w}$ one again has the opportunity to compute the gradient in parallel.

Online setting: Since the focus of the paper is on transitive rankings, we concentrate on online loss models that have more structure than just weighted sum of misordered pairs. The only such model that we are aware of assigns a linear cost over the assignment matrix of objects to that rank position (Helmbold and Warmuth, 2009), or their weighted analogue, doubly stochastic matrix that does a "soft matching". The most performant algorithm in this class is PermELearn (Helmbold and Warmuth, 2009). This algorithm's objective is to perform close to the best possible *fixed assign ment matrix*. Its cumulative complexity over T rounds of the algorithm is $\mathcal{O}(Td^6 \log(Td)$. For any large problem this is intractable because d is the size of the universe of all items to rank.

In comparison, our model can deal with varying set of items that need to be ordered in each round. The adversary provides the feature matrix $\boldsymbol{A}_t$ of d_t items that it has ranked at round t, but that order is not revealed untill the learner responds with a "scoring vector" $\boldsymbol{w}_t$. The learner is then charged a cost of $G_t(\boldsymbol{w}_t)$ as defined in (3.18) using a twice differentiable σ strongly convex function ϕ_t with L Lipschitz continuous gradient. The order and the function ϕ_t is then revealed for the learner. The objective is to minimize the cumulative loss $\sum_t G_t(\boldsymbol{w}_t)$. Here we will essentially plug in the known regret bound results obtained for online gradient descent for strongly convex, Lipschitz gradient functions (Hazan et al., 2007). For the t^{th} gradient update we use the t^{th} term of the gradient (3.19) with a learning rate of $\frac{1}{\sigma t}$ as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{1}{\sigma t}\nabla G_t(\{\boldsymbol{r}_t^*\}, \boldsymbol{w})$$

where $\boldsymbol{r}_t^* = \mathrm{Argmin}_{\boldsymbol{r}_t \in \mathcal{R}\downarrow \cap \mathcal{S}_t} G_i(\boldsymbol{r}_t, \boldsymbol{w})$.

Theorem 3. *(Hazan et al., 2007) The online gradient algorithm applied in an online setting to a s strongly function that has L Lipschitz continuous gradients has regret $\mathcal{O}(\frac{L^2}{\sigma}\log T)$.*

Neither the algorithm nor the bound is new, what is novel though is that the ranking problem of such combinatorial nature can be transformed into a form, without loss in generality, that this algorithm can exploit.

3.5 Experiments

We evaluated the ranking performance of the proposed margin equipped monotone retargeting (MEMR) approach on the benchmark LETOR 4.0 datasets (MQ2008) (Liu et al., 2007) as well as the OHSUMED dataset (Hersh et al., 1994). Each of these datasets is prepartitioned into five-fold validation sets for easy comparison across algorithms. We follow the experimental setup described in (Acharyya et al., 2012). The regularization parameter for the targets were set so that the marginal cost function was 0.001 strongly convex. The

MQ 2008 NDCG			
	I-div	SQ	KL
MEMR	0.7418	**0.7619**	0.7553
MR	0.7339	0.7398	0.7451
(Ravikumar et al., 2011)	0.5892	0.7344	0.7399

Table 3.5: Test NDCG on datasets MQ 2008.

OHSUMED NDCG			
	I-div	SQ	KL
MEMR	0.6983	**0.7250**	0.6944
MR	0.7000	0.6878	0.6997
(Ravikumar et al., 2011)	0.5805	0.6892	0.6947

Table 3.6: NDCG on OHSUMED dataset.

best model was identified as the model with highest NDCG (Järvelin and Kekäläinen, 2000) on the validation set.

The MR algorithm on which MEMR is based is our primary baseline. Recall that the MR algorithm has been shown to handsomely outperform many of the current state of the art techniques such as Listnet and RankCosine. For reference we also tabulate the results obtained by the state of the art NDCG consistent methods introduced by Ravikumar et. al (Ravikumar et al., 2011). We did not re-implement the MR family of algorithms but use the numbers reported in Acharyya et. al. including the baselines that they compared against.

The results are reported in tables 3.5 and 3.6. MEMR does indeed outperform MR, but this is not observed for all Bregman divergences. One prominent difference from the MR family is that square loss with MEMR does significantly better than square loss with MR. Our working hypothesis for the much improved behavior of square loss is that the simplex normalization used in MR artificially constraints the system from exploring regions of the parameter space with good test performance.

3.6 Conclusion

In this chapter we introduced a family of new cost functions for ranking. The cost function takes into account all possible monotonic transforms of the target scores, and we show how such a cost function can be optimized efficiently. Because the sole objective of learning to rank is to output good permutations on unseen data, it is desirable that the cost function be a function of such permutations. Though several permutation dependent cost functions have been proposed, they are extremely difficult to optimize over and one has to resort to surrogates and/or cut other corners. We show that with monotone retargeting with Bregman divergences such contortions are unnecessary. In addition, the proposed cost function and algorithms have very favorable statistical, optimization theoretic, as well as empirically observed properties. Other advantages include extensive parallelizability due to simple simultaneous projection updates that optimize a cost function that is convex not only in each of the arguments separately but also jointly under appropriate choice.

Chapter 4

Learning Bregman Divergences for Ranking

This chapter is concerned with prediction using generalized linear models with an unknown link function and is particularly suited for learning to rank. We begin with a motivating example, several of its assumptions will be relaxed later on.

Let a generalized linear relation $y_i = g(\langle \boldsymbol{u}, \boldsymbol{x}_i \rangle)$ hold with an unknown, continuously differentiable, strictly monotonic function $g(\cdot)$ and an unknown vector $\boldsymbol{u} \in \mathcal{W} \subset \mathbb{R}^n$, on the data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)_{i=1}^m\}$. We have to recover $\boldsymbol{u}$ and predict on future examples. The set $\mathcal{W}$ is a mechanism to control the complexity of the resulting predictor. It can be given explicitly, for example as an ℓ_1 or an ℓ_2 ball, or it can be given implicitly by a regularizing function that will be denoted by $\mathfrak{R}(\cdot)$. Although we motivate our cost function in terms of a perfect $\boldsymbol{u}$, no such vector need to exist, neither for the algorithms proposed nor for the analysis.

When $g(\cdot)$ is the identity function, the canonical technique is to minimize $||\boldsymbol{y} - X\boldsymbol{w}||^2$ with respect to $\boldsymbol{w} \in \mathcal{W}$. Iterative methods applied to this problem generate a sequence $\boldsymbol{w} \to \boldsymbol{w}_*$ that satisfies $\nabla_{\boldsymbol{w}=\boldsymbol{w}_*} ||\boldsymbol{y} - X\boldsymbol{w}||^2 \in -\mathcal{N}_{\mathcal{W}}(\boldsymbol{w}_*)$, where $\mathcal{N}_{\mathcal{W}}(\boldsymbol{w}_*)$ is a normal direction of the constraint set $\mathcal{W}$ at $\boldsymbol{w}_*$. Strict convexity of the cost in $\boldsymbol{w}$ ensures

$\boldsymbol{w}_* = \boldsymbol{u}$ if $\boldsymbol{u} \in \mathcal{W}$.

When $g(\cdot)$ is a known function, but not identity, the iterative technique of generating $\boldsymbol{w} \rightarrow \boldsymbol{w}_*$ that satisfies $\nabla_{\boldsymbol{w}=\boldsymbol{w}_*}||\boldsymbol{y} - \boldsymbol{g}(X\boldsymbol{w})||^2 \in -\mathcal{N}_{\mathcal{W}}(\boldsymbol{w}_*)$ loses its effectiveness in the general case. In this case $||\boldsymbol{y} - \boldsymbol{g}(X\boldsymbol{w})||^2$ need not be convex in $\boldsymbol{w}$ and may contain exponentially many (in dimensionality of $\boldsymbol{x}$) local minima (Auer et al., 1995). Without further assumptions it becomes impossible to restrict $||\boldsymbol{w}_* - \boldsymbol{u}||^2$ to an arbitrary low value, making recovery intractable.

An effective alternative, that applies to a known $g(\cdot)$, is to minimize a *matching* Bregman divergence (Auer et al., 1995). Recall that given a strictly convex, continuously differentiable function $\phi(\cdot)$ the corresponding Bregman divergence is $D_\phi\big(x\Big|\Big|y\big) = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y)\rangle$. If the relation $(\nabla\phi)^{-1}(\cdot) = g(\cdot)$, holds then the divergence $D_\phi\big(\boldsymbol{y}\Big|\Big|g(X\boldsymbol{w})\big)$ becomes convex in $\boldsymbol{w}$, strictly so if X has rank n (Auer et al., 1995). This ensures recovery, and the divergence in this case is said to "match" the transform $g(\cdot)$. Its minimizer is the maximum likelihood estimate of a canonical generalized linear model (GLM) (McCulloch and Searle, 2001) whose inverse link function is $g(\cdot) = (\nabla\phi)^{-1}(\cdot)$: a familiar object for statisticians and machine learners.

It should now be clear that the ability to recover $\boldsymbol{u}$ is affected by whether the loss function matches the transform $g(\cdot)$ or not. An explicit form of the function $g(\cdot)$ is often assumed for convenience, which in turn fixes the choice of the matching divergence. However, unless one has explicit control over the data generating process, $g(\cdot)$ is rarely known. Practioners typically assume a suitable or popular form of $g(\cdot)$ and proceed. Furthermore, the infinite cardinality of possible $g(\cdot)$'s rules out exhaustive hypothesis testing. Thus, there is a convincing case for *learning* the recovery-facilitating loss function when $g(\cdot)$ is unknown. This is the focus of this chapter. Given a strictly (or strongly) convex regularizer $\mathfrak{R}(\boldsymbol{w})$, a non-negative scalar c and $\boldsymbol{y}$ of dimensionality n, a candidate cost functional that

captures this notion is the following:

$$\min_{\boldsymbol{w},\phi(\cdot)\in\mathcal{C}} \frac{1}{n} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big) + c\,\mathfrak{R}(\boldsymbol{w}) \tag{4.1}$$

$$\equiv \min_{\substack{\boldsymbol{w},\phi(\cdot)\in\mathcal{C}\\ \boldsymbol{\theta}\in Range(X)}} \frac{1}{n} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}\left(\boldsymbol{\theta}\right)\big) + \inf_{X\boldsymbol{w}=\boldsymbol{\theta}} c\mathfrak{R}(\boldsymbol{w}). \tag{4.2}$$

The space $\mathcal{C}$ of all continuously differentiable strictly convex functions is convex. It is also infinite dimensional. In the absence of other simplifying restrictions, that we loathe to make, such as assuming a finite dimensional parameterization of a subset of it, or fitting $g(\cdot)$ with a spline and enforce monotonicity, this seems a challenging problem.

Close in intent and particularly notable is the paper by Kalai and Sastry (2009) where they propose the `isotron` algorithm that achieves a $\mathcal{O}(\frac{1}{T})$ bound on square loss $||\boldsymbol{y} - \boldsymbol{g}(X\boldsymbol{w})||^2$ (note, not on $||\boldsymbol{w}_* - \boldsymbol{u}||^2$) in spite of the non-convexities introduced by $g(\cdot)$. Reading the paper one readily appreciates how lack of convexity makes the analysis significantly more cumbersome. We believe that the approach proposed in this chapter is simpler, and under mild assumptions, the convergence rates are exponentially faster. This does not diminish the value of the paper (Kalai and Sastry, 2009), to the contrary it shows that non-convexity can at times be partially (if but painfully) conquered, and as we shall show for the `isotron` algorithm, by virtue of some hidden convexity.

Although developed independently, there are intriguing connections between the two approaches. We devote Section 4.6 to explain them. In retrospect, we note that an unintended consequence of our proposition has been that it sheds light on the question: how or why was it possible to conquer non-convexity in this particular case.

The **Learning to rank** problem provides another strong motivation for the cost function (4.1). Let $\{(\boldsymbol{x}_i, y_i)_{i=1}^m\}$ be drawn from a set $\mathcal{X}$ ordered by $y(\boldsymbol{x})$. We want to learn $\boldsymbol{u}$ such that the order induced by the $\langle u, \boldsymbol{x}\rangle$ suffers low permutational loss. The only loss function family, statistically consistent with the popular permutational loss: NDCG (Järvelin and Kekäläinen, 2000), is $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}\left(\langle\boldsymbol{w},\boldsymbol{x}\rangle\right)\big)$ (Ravikumar et al., 2011).

Structural risk minimization (Vapnik, 1998) then justifies minimizing the regularized empirical loss $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}\left(\langle\boldsymbol{w},\boldsymbol{x}\rangle\right)\big)$ over $\phi, \boldsymbol{w}$ to reduce expected loss in the future.

For the ranking case it is possible to push the model even further. Note that the predictions need not recover $\boldsymbol{y}$ pointwise to obtain the correct ranking. Predicting any monotonic transformation of $\boldsymbol{y}$ would be sufficient. This observation points to the following, natural modification of (4.1):

$$\min_{\boldsymbol{w},\phi(\cdot)\in\mathcal{C},\boldsymbol{z}\in\mathcal{R}\downarrow(\boldsymbol{y})} \frac{1}{n} D_{\boldsymbol{\phi}}\big(\boldsymbol{z}\big|\big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big) + c\,\mathfrak{R}(\boldsymbol{w})$$

where $\mathcal{R}\downarrow(\boldsymbol{y})$ is the set of vectors isotonic to $\boldsymbol{y}$.

Restricted Output Space: In prediction problems one often has some prior knowledge about output space, for example one might know that the outcome corresponding to an $\boldsymbol{x}$ is in some strict subset of $\mathbb{R}$. Indeed a common way to choose the link function of a canonical GLM is to choose the link function such that its domain matches the output space. For example to predict probabilities, the popular link function is log-odds whose domain is the interval [0,1]. This choice obtains the logistic regression model. Our framework can easily incorporate knowledge about the output space, in particular one may specify a convex subset of $\mathbb{R}$ (in other words an interval) to be the output space for $\boldsymbol{x}$, however the output space for X has to have a Cartesian product structure.

Notation: Vectors are denoted by bold lower case letters, matrices are capitalized. $||\boldsymbol{x}||$ denotes the ℓ_2 norm. The space of all strictly convex differentiable and separable functions is denoted by $\mathcal{C}$. When decorated with a superscript, e.g., $\mathcal{C}^s$ it denotes a subset consisting of all strongly convex functions, the superscript specifies the modulus. We use subscripts similarly for the subset of functions with Lipschitz continuous gradients e.g. $\mathcal{C}_l$. We use the wildcard symbol $\mathcal{C}\star$ to stand for one of $\mathcal{C}, \mathcal{C}_l, \mathcal{C}^s, \mathcal{C}^s_l$ when the discussion applies uniformly. The symbol $\mathcal{R}\downarrow \subset \mathbb{R}^n$ will denote a set of all vectors that are sorted by the component (it does not matter whether such vectors are sorted up or down, as long as that choice remains fixed).

Background: Convex duality, Bregman divergences and their relation to exponential family densities will play a major role in the chapter. Relevant results are summarized in Chapter 2. Recall that **Fenchel-Young Inequality** $\phi(\boldsymbol{y}) + \phi^*(\boldsymbol{\theta}) - \langle \boldsymbol{y}, \boldsymbol{\theta} \rangle \geq 0$ plays an important role in convex analysis (Rockafellar, 1996), and as we shall see, in this work as well.

The **infimal convolution** of $\boldsymbol{\phi}_1(\cdot)$ and $\boldsymbol{\phi}_2(\cdot)$ is denoted in this chapter by $\boldsymbol{\phi}_1 \oplus \boldsymbol{\phi}_2$ and is defined as: $[\boldsymbol{\phi}_1 \oplus \boldsymbol{\phi}_2](\boldsymbol{y}) = \inf_{\boldsymbol{x}} \boldsymbol{\phi}_1(\boldsymbol{x}) + \boldsymbol{\phi}_2(\boldsymbol{y} - \boldsymbol{x})$ (Rockafellar, 1996). The following identities will be useful:

$$[\alpha \boldsymbol{\phi}(\boldsymbol{\lambda})]^* = \alpha \boldsymbol{\phi}^*(\frac{\boldsymbol{x}}{\alpha}), \quad [\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2]^*(\cdot) = [\boldsymbol{\phi}_1^* \oplus \boldsymbol{\phi}_2^*](\cdot). \tag{4.3}$$

Recall that an **exponential family *density*** [1] of a random variable Y has the form $P(Y = \boldsymbol{y} \mid \boldsymbol{\theta}) = \exp^{\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - \psi(\boldsymbol{\theta})}$. These densities are indexed by its *natural* parameter $\boldsymbol{\theta}$. It is well known (Lehmann, 1983) that not only is the domain $\Theta = \left\{ \boldsymbol{\theta} \,\middle|\, \int_{\mathcal{Y}} \exp^{\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle} < \infty \right\}$ of the parameter a convex set, the normalizer $\psi(\boldsymbol{\theta})$, is also a convex function (strictly so if $\mathcal{Y}$ is affinely independent) called the log partition function. All moments of Y can be recovered from it, for example:

$$\mathbb{E}[Y] = \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) = (\nabla \phi)^{-1}(\boldsymbol{\theta}). \tag{4.4}$$

The log partition function $\psi(\cdot)$, its domain Θ, its Legendre dual $\phi(\cdot)$ which is the negative entropy of the random variable will all play an important role in the chapter.

Maximum likelihood obtains an estimate of $\boldsymbol{\theta}$ as the maximizer of the sample log likelihood: $\boldsymbol{\theta}^* = \operatorname{Argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{y} \mid \boldsymbol{\theta})$. For exponential family this is related to Bregman divergence as follows:

$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname{Argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{y} \mid \boldsymbol{\theta}) - \log P(\boldsymbol{y} \mid \boldsymbol{\theta}^*) = \operatorname{Argmin}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}^*) - \langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \boldsymbol{y} \rangle \\ &= \operatorname{Argmin}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) - \psi(\boldsymbol{\theta}^*) - \langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}^*) \rangle = \operatorname{Argmin}_{\boldsymbol{\theta}} D_\psi(\boldsymbol{\theta} || \boldsymbol{\theta}^*) = \operatorname{Argmin}_{\boldsymbol{\theta}} D_{\boldsymbol{\phi}}\big(\boldsymbol{y} \,\big|\big|\, (\nabla \phi)^{-1}(\boldsymbol{\theta})\big).\end{aligned}$$

Generalized linear models (GLM) assume a *specific* exponential family probability density

[1] with respect to a base measure. For notational simplicity the base measure will be omitted.

for Y conditioned on $\boldsymbol{x}$. In particular, the natural parameter $\boldsymbol{\theta}$ is assumed to be a linear function of $\boldsymbol{x} \in \mathcal{X}$. Note that choosing a particular exponential family is equivalent to choosing a particular convex function $\phi(\cdot)$. As can be seen from equation (4.4), the gradient of $\phi(\cdot)$ maps the expectation space into the natural parameter space and this mapping is called the *link function*. Estimating $\boldsymbol{\theta}$ using conditional maximum likelihood leads to

$$\boldsymbol{\theta}^* = \mathrm{Argmin}_{\boldsymbol{\theta}}\, D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}\left(\langle \boldsymbol{x}, \boldsymbol{w}\rangle\right)\big) = \mathrm{Argmin}_{\boldsymbol{\theta}}\, D_{\phi}\big(\boldsymbol{y}\big|\big|_{\boldsymbol{y}\sim\exp^{\langle\boldsymbol{\theta},\boldsymbol{y}\rangle-\phi^*(\boldsymbol{\theta})}} \mathbb{E}\,[\boldsymbol{y}]\big). \tag{4.5}$$

Thus the objective (4.1) *can be also seen as finding the member from the exponential family that fits the empirical conditional expectations* $\boldsymbol{y}$*, subject to regularization.*

4.1 Formulation

The key objects of our study are the properties of (4.1) and algorithms to minimize it. From equation (4.5), it should be clear that when $\phi(\cdot)$ is known, this is a well understood problem with existing and well vetted algorithms (McCulloch and Searle, 2001), (Pietra et al., 1997). The novelty is in optimizing over the infinite dimensional space of $\phi(\cdot)$. In light of this optimization, however, even equation (4.5) takes on new complexities. As we optimize iteratively over ϕ, we will not know the value of $\phi(\cdot)$ everywhere (after all we only have finitely many evaluations of its gradient), in fact we will not have any direct representation of $\phi(\cdot)$ at all, making evaluation of (4.5) impossible. The optimization algorithm has to deal with this.

A major source of complication and one of the reasons why formulation (4.1) cannot be trivially handled over to a standard convex optimization package is that ϕ is a function, hence infinite-dimensional. There are no basis set for such functions, making (linear) parameterization that is both complete and contained impossible.

The fact that $\phi(\cdot)$ couples the divergence as well a one of the arguments, is another significant impediment. It prevents us from exploiting a strikingly nice property of

Bregman divergences that the minimizer of some associated optimization problems become independent of the choice of the convex function used to define the divergence, a prototypical example is Proposition 1 of Banerjee et al. (2005). The following result obtained in our prior work (Acharyya et al., 2012) comes closest to our current need:

Lemma 12. *(Acharyya et al., 2012) If the Bregman divergence $D_\phi\big(\cdot\big|\big|\cdot\big)$ is separable, and $\mathcal{R}\downarrow$ the set of vectors $\boldsymbol{y}$ in $\mathbb{R}^n$ that are in sorted order, that is, $v_i < v_j$ if $i < j$ then the minimizer $\operatorname{Argmin}_{\boldsymbol{y}\in\mathcal{R}\downarrow} D_\phi\big(\boldsymbol{x}\big|\big|\boldsymbol{y}\big)$ is independent of ϕ for all $\boldsymbol{x} \in \operatorname{dom}\phi(\cdot)$.*

Unfortunately these results are for the uncoupled case and cannot be used directly. So in what follows, we have to overcome: (i) infinite dimensionality and (ii) coupling. We will, however, make use of the following property although somewhat indirectly.

Corollary 2. *Let A be a symmetric positive definite matrix that defines the squared Mahalonobis distance, the minimizer $\operatorname{Argmin}_{\boldsymbol{y}\in\mathcal{R}\downarrow} ||\boldsymbol{x}-\boldsymbol{y}||_A^2$, is independent of the choice of A if it diagonal.*

Proof. Squared Mahalonobis distance $||(\boldsymbol{x})-(\boldsymbol{y})||_A^2$ is a Bregman divergence and separable when A is diagonal. □

4.1.1 Uniqueness of the Minimum

For a fixed, strictly convex ϕ, equation (4.5) has an unique optimum because (4.5) is strictly convex. In formulation (4.1) both $\boldsymbol{w}$ and $\phi(\cdot)$ vary, so it is important to know whether the joint optima is unique. We show that $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$ is jointly convex in the function $\phi(\cdot)$ and vector $\boldsymbol{w}$. Thus with a strictly convex regularizer $\mathfrak{R}(\boldsymbol{w})$ the optimum is unique in $\boldsymbol{w}$.

Theorem 4. *If $\phi \in \mathcal{C}$ then the functional $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$ is jointly convex in $\phi, \boldsymbol{w}$.*

Proof. Let $\boldsymbol{\theta} = \langle \boldsymbol{x}, \boldsymbol{w}\rangle$ and $\bar{\boldsymbol{\theta}} = \alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2$. It will then be sufficient to show that $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$ is convex in $g(\cdot)$ and $\boldsymbol{\theta}$. Recall $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big) = \phi(\boldsymbol{y})+\psi(\boldsymbol{\theta})-$

$\langle \boldsymbol{y}, \boldsymbol{\theta} \rangle$ is the Fenchel-Young gap: $\phi(\boldsymbol{y}) + \phi^*(\boldsymbol{\theta}) - \langle \boldsymbol{y}, \boldsymbol{\theta} \rangle$ defined in Chapter 2 and denoted here by $F\left(\begin{smallmatrix} \phi \\ \boldsymbol{\theta} \end{smallmatrix}\right)$.

Showing joint convexity is equivalent to showing $\overbrace{\alpha F\left(\begin{smallmatrix} \phi_1 \\ \boldsymbol{\theta}_1 \end{smallmatrix}\right) + (1-\alpha) F\left(\begin{smallmatrix} \phi_2 \\ \boldsymbol{\theta}_2 \end{smallmatrix}\right)}^{A} \geq \overbrace{F\left(\begin{smallmatrix} \alpha\left(\begin{smallmatrix} \phi_1 \\ \boldsymbol{\theta}_1 \end{smallmatrix}\right) + \\ \left((1-\alpha)\begin{smallmatrix} \phi_2 \\ \boldsymbol{\theta}_2 \end{smallmatrix}\right) \end{smallmatrix}\right)}^{B}$.

$$
\begin{aligned}
A &= [\alpha\phi_1 + (1-\alpha)\phi_2]\Big(\boldsymbol{y}\Big) + \alpha\psi(\boldsymbol{\theta}_1) + (1-\alpha)\psi(\theta_2) - \left\langle \boldsymbol{y}, \bar{\boldsymbol{\theta}} \right\rangle. \\
B &= [\alpha\phi_1 + (1-\alpha)\phi_2]\Big(\boldsymbol{y}\Big) - \left\langle \boldsymbol{y}, \bar{\boldsymbol{\theta}} \right\rangle + [\alpha\phi_1 + (1-\alpha)\phi_2]^*\Big(\bar{\boldsymbol{\theta}}\Big).
\end{aligned}
$$

$$
\begin{aligned}
A - B &= \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - [\alpha\phi_1 + (1-\alpha)\phi_2]^*(\bar{\boldsymbol{\theta}}) \\
&= \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - [(\alpha\phi_1)^* \oplus ((1-\alpha)\phi_2)^*](\bar{\boldsymbol{\theta}}) \\
&= \alpha\phi_1^*(\boldsymbol{\theta}_1) + (1-\alpha)\phi_2^*(\boldsymbol{\theta}_2) - \\
&\quad \Big[\min_z (\alpha\phi_1)^*(z) + ((1-\alpha)\phi_2)^*(\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2 - z)\Big] \\
&\geq 0, \quad \text{obtained by setting } z = \alpha\boldsymbol{\theta}_1
\end{aligned}
$$

□

Corollary 3. *If $\phi(\cdot)$ is convex and $\mathfrak{R}(\boldsymbol{w})$ is strictly(strongly) convex then the cost function* (4.1) $\inf_\phi D_{\boldsymbol{\phi}}\Big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w})\Big) + c\mathfrak{R}(\boldsymbol{w})$ *is strictly(strongly) convex in $\boldsymbol{w}$.*

Using equation (2.4) $D_{\boldsymbol{\phi}}\Big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w})\Big)$ can be represented in terms of the function ϕ^* as $D_{\boldsymbol{\psi}}\Big(X\boldsymbol{w}\Big|\Big|\nabla\phi(\boldsymbol{y})\Big)$. Obviously, the cost function continues to enjoy the uniqueness of the minimum, but what is interesting is whether it is also jointly convex in this representation.

Theorem 5. *If $\phi^* \in \mathcal{C}$ then $D_{\boldsymbol{\psi}}\Big(X\boldsymbol{w}\Big|\Big|\nabla\phi(\boldsymbol{y})\Big)$ is jointly convex over ϕ^* and $\boldsymbol{w}$.*

Proof. Follows from a similar sequence of arguments as used in Theorem 4. □

Fenchel-Young Divergence: It should be evident from the proof of Theorem 4 that using the Fenchel-Young gap form $\phi(\boldsymbol{y}) + \phi^*(X\boldsymbol{w}) - \langle \boldsymbol{y}, X\boldsymbol{w} \rangle$, instead of the divergence form $D_{\boldsymbol{\phi}}\big(\boldsymbol{y} \Big|\Big| (\nabla\phi)^{-1}(X\boldsymbol{w})\big)$ gets rid of the coupling present in the divergence form. The values computed by both the forms are of course equal when both are well defined. We now argue that the Fenchel-Young gap representation is to be preferred because it widens the scope of the formulation from differentiable convex functions to closed convex functions.

At the (at most finitely many) points where a closed convex function $\phi(\cdot)$ is not differentiable, the expression for the Bregman divergence becomes ambiguous. There are not one, but many "gradient" like (lower bounding) functions defined at such points, called subgradients. One among them needs to be chosen to evaluate the expression $\phi(\boldsymbol{y}) - \phi(\boldsymbol{x}) - \langle \boldsymbol{y} - \boldsymbol{x}, \partial(\boldsymbol{x}) \rangle$. Some such choices are $\sup_{\partial(\boldsymbol{x})} \langle \boldsymbol{y} - \boldsymbol{x}, \partial(\boldsymbol{x}) \rangle$, $\inf_{\partial(\boldsymbol{x})} \langle \boldsymbol{y} - \boldsymbol{x}, \partial(\boldsymbol{x}) \rangle$ (Kiwiel, 1988).

For our purposes, however, this ambiguity is artificial. Note that $\boldsymbol{y}$ lives in the domain of $\phi(\cdot)$ whereas $\boldsymbol{\theta} = X\boldsymbol{w}$ lives in the domain of the dual, $\phi^*(\cdot)$. The function $(\nabla\phi)^{-1}$ was enlisted to bring $\boldsymbol{\theta}$ into the domain of ϕ so that the divergence could be evaluated. However, using the Fenchel-Young gap form one can evaluate the same divergence directly, without the need for a mapping, which as we have shown may cease to be unique (or even exist) at certain points in the domain.

The Fenchel-Young gap form has been called *generalized Bregman divergence* in literature (Gordon, 1999), however since the same term has also been used to describe $\phi(\boldsymbol{y}) - \phi(\boldsymbol{x}) - \sup_{\partial(\boldsymbol{x})} \langle \boldsymbol{y} - \boldsymbol{x}, \partial(\boldsymbol{x}) \rangle$ we prefer the more explicit name Fenchel-Young divergence.

4.1.2 Role of Curvature and Smoothness of the Divergence

Let us denote $(\nabla\phi(\boldsymbol{\theta}))^{-1}$ by $\boldsymbol{s}(\boldsymbol{\theta})$ and a small positive number by ϵ. A scenario that we must avoid is the following: $||\boldsymbol{y} - \boldsymbol{s}(\boldsymbol{\theta}_t)|| \to \frac{1}{\epsilon} > 0$ yet $D_{\phi_t}(\boldsymbol{y}||\boldsymbol{s}(\boldsymbol{\theta}_t)) \to 0$. The limiting $\phi(\cdot)$ and $\boldsymbol{w}$ so obtained would be useless as devices of prediction or recovery. Bregman

divergence being the "excess" of a convex function over its local linear approximation, it is possible to reduce the divergence between two distant points by making the convex function approach linearity in between. Let us examine the nature of the degeneracy by considering a sequence of functions

$$\lim_t \phi_t(y) \to ay + c.$$

In this case the limiting Fenchel-Young divergence is given by

$$\lim_t \phi_t(y) + \phi_t^*(\theta) - \theta y \to \begin{cases} 0 \text{ if } \theta = a \\ \\ \infty \text{ otherwise} \end{cases}.$$

Thus our cost function may approach zero even if $||\boldsymbol{y} - \boldsymbol{s}(\boldsymbol{\theta}_t)|| \to \frac{1}{\epsilon} > 0$. This can be achieved by setting $\lim_t \phi_t(y) \to \theta_i y + c$, in the interval $[y_i, s(\theta_i))$ for all i. Note, however, that this cannot done arbitrarily. Convex functions are restricted to have monotone increasing gradients, hence the degenerate situation is possible only when $\boldsymbol{\theta}$ and $\boldsymbol{y}$ are in the same order. Thus as long as the components of $\boldsymbol{\theta}$ are distinct, this degeneracy is not a problem in case of a ranking application, because we want the cost to be zero when $\boldsymbol{\theta}$ and $\boldsymbol{y}$ are in the same order. However, it must be ensured that $\boldsymbol{\theta}$ does not converge to a vector $c\mathbf{1}$. For this we would require a data dependent condition that $\min_{\boldsymbol{v} \in Range(X), \boldsymbol{t} \in c\mathbf{1}} ||\boldsymbol{v} - \boldsymbol{t}|| > \frac{1}{\epsilon}$.

Restricting the $\phi(\cdot)$ optimization in (4.1) to a subset of $\mathcal{C}$ with a minimum, non-zero curvature clearly prevents such a degeneracy. This subset is denoted by $\mathcal{C}^s$ and is the set of $s-$strongly convex functions.

Enforcing curvature has the following additional benefits: (i) Strong convexity in $\phi(\cdot)$ (equivalently Lipschitz continuity in $(\nabla\phi)^{-1}(\cdot)$) facilitates prediction. Without further assumptions the function $(\nabla\phi)^{-1}$ can at best be known at finitely many points. Curvature and Lipschitz continuity allow one to make principled extrapolation outside of those points. (ii) Smoothness and curvature play an important role in yielding faster convergence rates of the optimization algorithms as shown in Table 4.1. The function $\phi(\cdot)$ is irrelevant to the rank

$\mathcal{C}\star$	$\mathfrak{R}(\boldsymbol{w})$	Convergence Rate	Algorithm
Convex	Strictly Convex	$\frac{1}{\sqrt{T}}$	Gradient Descent (GD)
Convex	Strongly Convex	$\frac{1}{T}$	Accelerated GD
Convex and smooth	Strictly Convex	$\frac{1}{T^2}$	Accelerated GD
Convex and smooth	Strongly Convex and smooth	$\exp(-\lambda T)$	Accelerated GD

Table 4.1: Convergence rates of gradient descent based algorithms

order and hence plays a lesser role in making rank predictions. However, strong convexity controls the 'learning capacity' of the function and directly affects its generalization.

Usually, constrained optimization is more time consuming than unconstrained and therefore one might anticipate that restricting the curvature of $\phi(\cdot)$ in optimization (4.1) comes at a higher computational burden. However, not only is there no extra computational burden, the presence of curvature gives rise to very fast convergence, summarized in Table 4.1.

Total and Uniform Convexity: As convenient as curvature restriction is, there is no denying that it rules out many continuously differentiable strictly convex functions, for example $\log(\sum \exp(x_i))$. This begs the question can the uniform curvature restriction be relaxed. Indeed, the weakest restrictions under which this is possible, without making assumption on the data X is that $\phi(\cdot)$ belongs to the class

$$\left\{ \phi | \delta\left(\frac{1}{\epsilon}\right) = \inf_{||\boldsymbol{x}-\boldsymbol{s}(\boldsymbol{\theta})|| \geq \frac{1}{\epsilon} > 0} D_\phi\big(\boldsymbol{y} \Big|\Big| \boldsymbol{s}(\boldsymbol{\theta})\big) > 0 \right\}.$$

For reasons of convenience, we work with a slightly stronger, sufficient class called *uniformly strictly convex*, these are functions that have a modulus of *uniformly strictly convexity* strictly greater than 0 . The modulus of *uniformly strictly convex* is defined as follows

$$\delta(\frac{1}{\epsilon}) = \inf_{||\boldsymbol{y}-\boldsymbol{x}|| > \frac{1}{\epsilon},\ \alpha \in [0,1]} \frac{\alpha\phi(\boldsymbol{x}) + (1-\alpha)\phi(\boldsymbol{y}) - \phi(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y})}{\alpha(1-\alpha)}.$$

Unlike modulus of strong convexity which is a number, the modulus of *uniformly strict convexity* of $\phi(\cdot)$, is a function of ϵ. For any convex function $\phi(\cdot)$ this modulus is (i) non-

decreasing, (ii) $o(||\epsilon||^d)$ for some $d \geq 0$ i.e. as $e \to 0$ $\delta(e) \to 0$. It can further be shown that $\frac{\delta(t)}{t}$ is non decreasing. We recover $s-$strong convexity by choosing $\delta(||\boldsymbol{y} - \boldsymbol{s}||) = s||\boldsymbol{y} - \boldsymbol{s}||^2$. For an application where we do not want to be restricted to strongly convex functions alone, one can choose an appropriate $\delta(\cdot)$ and restrict the formulation (4.1) to such *uniformly strictly convex* functions. The modulus quantifies the trade off between the distance $||\boldsymbol{y} - \boldsymbol{s}||$ and how small can the Bregman divergence be and further, satisfies the properties of non-decrease and $o(||\epsilon||^d)$ for some $d \geq 0$.

For convenience we shall further impose that the modulus for $\phi(\cdot)$ satisfies $\delta(||\boldsymbol{y} - \boldsymbol{s}||) = s^{\gamma-1}||\boldsymbol{y} - \boldsymbol{s}||^\gamma$ for $\gamma \geq 2$. For such a function $\phi(\cdot)$ we obtain the following inequalities

$$\delta(||\boldsymbol{y} - \boldsymbol{s}||) \leq \langle \nabla\phi(\boldsymbol{y}) - \nabla\phi(\boldsymbol{s}), \boldsymbol{y} - \boldsymbol{s} \rangle \leq ||\nabla\phi(\boldsymbol{y}) - \nabla\phi(\boldsymbol{s})||_*||\boldsymbol{y} - \boldsymbol{s}|| \tag{4.6}$$

where $||\cdot||_*$ is the dual norm of $||\cdot||$. Using $\delta(||\boldsymbol{y} - \boldsymbol{s}||) = s^{\gamma-1}||\boldsymbol{y} - \boldsymbol{s}||^\gamma$ we obtain that $||\nabla\phi(\boldsymbol{y}) - \nabla\phi(\boldsymbol{s})||_* \geq s||\boldsymbol{s} - \boldsymbol{y}||^{\gamma-1}$ and therefore

$$||(\nabla\phi)^{-1}(\boldsymbol{w}) - (\nabla\phi)^{-1}(\boldsymbol{v})|| \leq \frac{1}{s}||\boldsymbol{w} - \boldsymbol{v}||_*^{\frac{1}{\gamma-1}} \tag{4.7}$$

in other words $(\nabla\phi)^{-1}(\cdot)$ is $(\frac{1}{s}, \frac{1}{\gamma-1})$ Holder continuous. This is important because the gradient of the cost function $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$ with respect to $\boldsymbol{\theta}$ and evaluated at some particular $\phi(\cdot)$ has the same smoothness coefficient as $(\nabla\phi)^{-1}(\cdot)$.

Holder continuity of the gradient will be beneficial because gradient based algorithms that are optimal in the first order oracle model for convex functions with (L_ν, ν) Holder continuous gradients are known (Nesterov, 2013). They achieve a rate of $T^{-\frac{1+3\nu}{2}}$. Quite remarkably the *accelerated gradient method* (see Section 4.2) that we recommend for the case that $\phi(\cdot)$ is $s-$strongly convex, can be re-used for the Holder continuous gradient case and still achieve the optimal rate (Devolder et al., 2011), provided an adjusted, effective Lipschitz constant is used.

Although we can handle *uniformly strictly convex* functions, we emphasize that the associated convergence rates are slower. Unless there are compelling reasons to consider a class beyond strongly convex functions there are little justification for opting for a slower method.

4.2 Optimization

Convex Marginal Function:

In the forthcoming analysis a prominent role will be played by the convex marginal functions $\inf_{\phi \in \mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$, they will be collectively denoted by $m\star(\boldsymbol{w})$. Note that $m\star(\boldsymbol{w})$ is a function of $\boldsymbol{w}$ alone. It follows from joint convexity (established in Theorem 4) that the marginals are convex, but do they also inherit smoothness of gradients ? In the next Lemma we establish that if we minimize over convex functions $\phi(\cdot)$ for which $(\nabla)^{-1}\phi(\cdot)$ is l Lipschitz continuous, this property continues to hold for the marginal.

Lemma 13. *If $\phi(\cdot)$ is $\frac{1}{l}$ strongly convex, then the convex marginal function $m_l(\boldsymbol{w}) = \inf_{\phi \in \mathcal{C}_l} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$ has a gradient with Lipschitz constant at least l.*

Proof. Let $\tilde{\phi} \in \operatorname{Argmin}_{\phi \in \mathcal{C}_l} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w}_1)\big)$. Then $D_{\tilde{\phi}}\Big(\boldsymbol{y}||(\nabla\tilde{\phi})^{-1}(X\boldsymbol{w})\Big)$ is a tight upper bound of $m_l(\boldsymbol{w})$ with the same gradient at $X\boldsymbol{w}_1$. Since $\tilde{\phi} \in \mathcal{C}_l$, the upper bound has l Lipschitz gradient, therefore gradient of $m_l(\boldsymbol{w})$ has a Lipschitz constant at least l. □

An optimization technique that is very popular in machine learning when there are two or more sets of variables that need to be optimized over, is block coordinate descent (Tseng, 2001). However in our setting, naive block coordinate minimization over $\boldsymbol{w}$ and $\boldsymbol{\phi}$ does not readily apply. First of all, it is not clear how one may optimize over the space of functions $\mathcal{C}\star$ without parameterization. Secondly, even if one could optimize over the infinite dimensional set $\mathcal{C}\star$, for a fixed $\boldsymbol{w}$, the optimizing *function* need not be unique because $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$ is only convex in $\phi(\cdot)$ and not strictly so. This is problematic

Gradient Descent (Nemirovski, 2001)	Accelerated Gradient Descent (Nemirovski, 2001)
Input: $\nabla m\star\,(\cdot), a, b$ Initialize $\boldsymbol{w}^0, t = 0$. **repeat** $\boldsymbol{w}^{t+1} = w^t - \frac{a}{b+\sqrt{t}}\nabla m\star\,(\boldsymbol{w}^t)$ **until** Converged	**Input:** $\nabla m\star\,(\cdot)$, Lipschitz constant l Initialize $\boldsymbol{w}^0, a^0 = 1, t = 0$. **repeat** $\boldsymbol{x}^t = \boldsymbol{w}^t - \frac{1}{l}\nabla m\star\,(\boldsymbol{w}^t)$ $a^{t+1} = \frac{(1+\sqrt{4(a^t)^2+1})}{2}$ $\boldsymbol{w}^{t+1} = \boldsymbol{x}_i + \frac{a^t-1}{a^{t+1}}(\boldsymbol{x}^t - \boldsymbol{x}^{t-1})$ **until** Converged

Table 4.2: Accelerated and (un-accelerated) Gradient Descent

because in absence of other assumptions, unique attainment of block-wise minimum is required for convergence of block coordinate descent (Bertsekas, 1999). In our case even the otherwise standard optimization over the $\boldsymbol{w}$ block requires special consideration because we cannot evaluate the cost function. This is so because the function $\phi(\cdot)$ will neither be known in closed form, nor everywhere.

On the other hand if we could compute the gradient of $m\star\,(\boldsymbol{w}) = \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}\,(X\boldsymbol{w})\big)$ and minimize $m\star\,(\boldsymbol{w}) + \mathfrak{R}(\boldsymbol{w})$ with this information, we would have achieved objective (4.1). This is the strategy we adopt. The novelty primarily lies in constructing an efficient computational scheme to obtain the gradient. The proposed gradient computation scheme will be referred to as `GradMaPr` . We shall soon see that its time complexity is at most $\log$ factor worse than computing the gradient of the GLM loglikelihood with a *known* $g(\cdot)$. The gradient, once computed, will be used in an optimization algorithm that is optimal in the black-box first order oracle sense (Nemirovski, 2001) exploiting smoothness properties that the gradient may have. The only concern in the latter part is that the optimization algorithm that uses the gradient must not require function evaluation. Now we state the kind of rates that could be achieved with an optimal gradient based method, assuming that we would be successful in computing the gradient of $m\star\,(\boldsymbol{w})$.

Gradient descent (Table 4.2 left) optimizes $m(\boldsymbol{w}) + \mathfrak{R}(\boldsymbol{w})$ s.t. $\phi \in \mathcal{C}$ such that sub-optimality

$$m(\boldsymbol{w}^t) + \mathfrak{R}(\boldsymbol{w}^t) - \inf_{\boldsymbol{u}}[m(\boldsymbol{u}) + \mathfrak{R}(\boldsymbol{u})] \leq \mathcal{O}(\frac{1}{\sqrt{t}}).$$

Accelerated gradient descent (Table 4.2 right) optimizes $m(\boldsymbol{w}) + \mathfrak{R}(\boldsymbol{w})$ s.t. $\phi \in \mathcal{C}_l$ such that sub-optimality

$$m(\boldsymbol{w}^t) + \mathfrak{R}(\boldsymbol{w}^t) - \inf_{\boldsymbol{u}}[m(\boldsymbol{u}) + \mathfrak{R}(\boldsymbol{u})] \leq \mathcal{O}(\frac{1}{t^2}).$$

Note that wherever the algorithms in Table 4.2 require the gradient, a call to the function `GradMaPr` will be made. Setting aside the details of `GradMaPr` that we shall describe shortly, the Table 4.2 shows the complete algorithms for optimizing our cost function (4.1).

4.2.1 `GradMaPr` : Gradients by Marginalization and Projection

If one can compute a (sub)gradient of $\inf_{\phi \in \mathcal{C}_\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$, one can optimize the functional (4.1). Computing the (sub)gradient is the goal of this section. For ease of reference we will call the proposed gradient computation method `GradMaPr` . What one does with a (sub)gradient once computed is a concern separated from `GradMaPr` itself, and any optimization algorithm that can work without function evaluation, or any variational inequality solver will suffice. The key here is to tackle the infinite dimensionality of $\phi(\cdot)$. Accomplishing this efficiently, and without loss of generality is one of the key contributions of the chapter.

A striking feature of `GradMaPr` is that, in spite of the infinite dimensional structure, the time complexity of computing the gradient is at most a $\log$ factor worse than the GLM case: the linear in the dimension of $\boldsymbol{w}$ whereas for `GradMaPr` the complexity is $\mathcal{O}(d \log d)$. In terms of time complexity, the added generality obtained over a fixed GLM by virtue of searching over all possible convex functions comes at minimal extra cost.

Recall that the sets $\mathcal{C}, \mathcal{C}_l, \mathcal{C}^s, \mathcal{C}_l^s$ are all closed. This follows because the limit of a sequence of convex (alternatively, convex with Lipschitz gradient, strongly convex, strongly convex with Lipschitz gradients) functions is a convex (alternatively, convex with Lipschitz gradient, strongly convex, strongly convex with Lipschitz gradients) function. Thus we can

replace $\inf$ by $\min$ in the expression $\inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)$. Using subdifferential calculus (Rockafellar, 1996) we obtain

$$\partial_{\boldsymbol{\theta}}\min_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big) \in \underset{\phi_*\in\operatorname{Argmin}_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right)}{\operatorname{ConvHull}} \{(\nabla\phi_*)^{-1}(\boldsymbol{\theta})-\boldsymbol{y}\}. \tag{4.8}$$

To realize equation (4.8) word for word in an algorithm would entail computing the set $\{\phi_*\}=\operatorname{Argmin}_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$ first and then the subgradient from it. However it is the first step that is problematic because it involves an infinite dimensional optimization over the space of functions. The remaining of this section is about how to circumvent this.

Circumventing the Computation of ϕ_*

In the forthcoming analysis, an important role will be played by the following range sets

$$\mathcal{S}(\boldsymbol{\theta}) \triangleq \{\boldsymbol{s}|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta}),\phi\in\mathcal{C}\},\quad \mathcal{S}_l(\boldsymbol{\theta}) \triangleq \{\boldsymbol{s}|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta}),\phi\in\mathcal{C}_l\},$$
$$\mathcal{S}^s(\boldsymbol{\theta}) \triangleq \{\boldsymbol{s}|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta}),\phi\in\mathcal{C}^s\},\quad \mathcal{S}^s_l(\boldsymbol{\theta}) \triangleq \{\boldsymbol{s}|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta}),\phi\in\mathcal{C}^s_l\}.$$

They will be collectively denoted by $\mathcal{S}\star$ when smoothness and/or strong convexity is not important to the discussion.

A vector $\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})$ is in correspondence with each $\phi\in\mathcal{C}\star$ that satisfies $\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})$. Each such $\phi(\cdot)$ incurs a cost $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(\boldsymbol{\theta})\big)$. We define the function[2] $M\star(\boldsymbol{s},\boldsymbol{\theta})$ using their minimum

$$M\star(\boldsymbol{s},\boldsymbol{\theta}) \triangleq \min_{\phi\in\mathcal{C}\star|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})} D_{\phi}\big(\boldsymbol{y}\big|\big|\boldsymbol{s}\big) = \min_{\phi^*\in\mathcal{C}\star|\boldsymbol{s}=\nabla\phi^*(\boldsymbol{\theta})} D_{\boldsymbol{\psi}}\big(\boldsymbol{\theta}\big|\big|\nabla\phi(\boldsymbol{y})\big) \text{ using (2.4)} \tag{4.9}$$

[2]This defines all the variants $M(\boldsymbol{s},\boldsymbol{\theta}), M_L(\boldsymbol{s},\boldsymbol{\theta}), M^s(\boldsymbol{s},\boldsymbol{\theta}), M^s_L(\boldsymbol{s},\boldsymbol{\theta})$ and the wildcard $M\star(\boldsymbol{s},\boldsymbol{\theta})$.

Now, note that our original objective (4.1) is equivalent to minimizing

$$\min_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta}),\boldsymbol{\theta},X\boldsymbol{w}=\boldsymbol{\theta}} M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)+\inf_{X\boldsymbol{w}=\boldsymbol{\theta}} c\Re(\boldsymbol{w}). \tag{4.10}$$

What the reformulation (4.10) achieves is that now we have a finite dimensional optimization problem over $\mathcal{S}\star$. that is equivalent to the infinite dimensional optimization (4.1). Although the function $\phi(\cdot)$ does not occur in the cost (4.10) any more we still have not circumvented the computation of ϕ_* because it is needed to evaluate the function $M\star(\boldsymbol{s},\boldsymbol{\theta})$. However, let us establish some useful properties of $M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)$.

Theorem 6. *The function $M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)$ is convex in $\boldsymbol{s}\in\mathcal{S}\star$*

Proof. Consider two points $\boldsymbol{s}_1,\boldsymbol{s}_2$. For a fixed $\boldsymbol{\theta}$, each correspond to functions ϕ_1^* and ϕ_2^* that achieves the minimum as indicated in (4.9), incurring the cost $D_{\boldsymbol{\psi}}\left(\boldsymbol{\theta}\middle|\middle|\nabla\phi\left(\boldsymbol{y}\right)\right)$ with the respective functions. Now consider the point $\alpha\boldsymbol{s}_1+(1-\alpha)\boldsymbol{s}_2=\alpha\nabla\phi_1^*(\boldsymbol{\theta})+(1-\alpha)\nabla\phi_1^*(\boldsymbol{\theta})$ where $\alpha\in[0,1]$. It is clear that it corresponds to the function $\alpha\phi_1^*+(1-\alpha)\phi_2^*$. The cost function $D_{\boldsymbol{\psi}}\left(\boldsymbol{\theta}\middle|\middle|\nabla\phi\left(\boldsymbol{y}\right)\right)$ has already been proved to be jointly convex 5. □

Optimizing $M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)$: The (sub)gradient of $M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)$ can be computed by differentiating (4.9) and is obtained as follows:

$$\begin{aligned}\partial_{\boldsymbol{s}}M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right) &= \underset{\phi_*^*\in\mathrm{Argmin}_{\phi^*\in\mathcal{C}\star|\boldsymbol{s}=\nabla\phi^*(\boldsymbol{\theta})}D_{\boldsymbol{\psi}}\left(\boldsymbol{\theta}\middle|\middle|\nabla\phi(\boldsymbol{y})\right)}{\mathrm{ConvHull}} ([\nabla^2\phi_*^*])^{-1}(\nabla\phi_*^*(\boldsymbol{\theta})-\boldsymbol{y}) \\ &= \underset{\phi_*\in\mathrm{Argmin}_{\phi\in\mathcal{C}\star|\boldsymbol{s}=(\nabla\phi)^{-1}(\boldsymbol{\theta})}D_{\phi}\left(\boldsymbol{y}\middle|\middle|\boldsymbol{s}\right)}{\mathrm{ConvHull}} [\nabla^2\phi_*]_{(\boldsymbol{\theta}=\nabla\phi(\boldsymbol{s}))}(\boldsymbol{s}-\boldsymbol{y}).\end{aligned} \tag{4.11}$$

The Hessian $[\nabla^2\phi^*]$ is a diagonal positive definite matrix since ϕ^* is separable and convex. The derivative of $M\star\left(\boldsymbol{s},\boldsymbol{\theta}\right)$ w.r.t $\boldsymbol{w}$ is obtained similarly as

$$\partial_{\boldsymbol{w}}M\star(\boldsymbol{s},\boldsymbol{\theta})=X^{\dagger}\underset{\phi_*^*\in\mathrm{Argmin}_{\phi^*\in\mathcal{C}\star|\boldsymbol{s}=\nabla\phi^*(\boldsymbol{\theta})}D_{\boldsymbol{\psi}}\left(\boldsymbol{\theta}\middle|\middle|\nabla\phi(\boldsymbol{y})\right)}{[\nabla^2\phi_*^*]}\partial_{\boldsymbol{s}}M\star(\boldsymbol{s},\boldsymbol{\theta})=X^{\dagger}(\boldsymbol{s}-\boldsymbol{y}). \tag{4.12}$$

In (4.10) we have recast (4.1) as a regularized optimization featuring $M\star\,(\boldsymbol{s},\boldsymbol{\theta})$, to which, it seems, we could apply (sub)gradient descent in the joint space $(\boldsymbol{s},\boldsymbol{w})$ using (4.11) and (4.12). Even if we could, this is strongly discouraged because the components of the gradient is clearly linearly dependent. Observe, however, that we still do not have a computational scheme to identify $\phi(\cdot)^*_*$ that is required to compute $\partial_{\boldsymbol{s}} M\star\,(\boldsymbol{s},\boldsymbol{\theta})$ numerically.

Descending along Marginalized $M\star\,(\boldsymbol{s},\boldsymbol{\theta})$:

An alternative approach that is worth exploring is to use an optimal descent method with respect to $\boldsymbol{w}$ on the marginal function $\min_{\boldsymbol{s}} M\star\,(\boldsymbol{s},\boldsymbol{\theta})$ using its gradient, that is, we short circuit gradient descent steps on $\boldsymbol{s}$ by minimizing it fully for a given $\boldsymbol{w}$ and then take a gradient step along $\boldsymbol{w}$, potentially saving several intermediate steps. Recall that $M\star\,(\boldsymbol{s},\boldsymbol{\theta})$ itself involves a conceptual optimization over $\phi \in \mathcal{C}\star$, and now we have to minimize it further over $\boldsymbol{s}$ to obtain $\boldsymbol{s}_*(\boldsymbol{\theta}) = \operatorname{Argmin}_{\boldsymbol{s}} M\star\,(\boldsymbol{s},\boldsymbol{\theta})$.

If we could carry out the minimization over $\boldsymbol{s}$, the subgradient of the marginal would be:

$$\partial_{\boldsymbol{w}} \inf_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star\,(\boldsymbol{s},\boldsymbol{\theta}) = X^\dagger \partial_{\boldsymbol{\theta}} \inf_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star\,(\boldsymbol{s},\boldsymbol{\theta}) = \operatorname{ConvHull}_{\boldsymbol{s}_*(\boldsymbol{\theta})} X^\dagger(\boldsymbol{s}_*(\boldsymbol{\theta}) - \boldsymbol{y}). \quad (4.13)$$

Perhaps surprisingly, as we shall show soon (Theorem 7), not only is $\boldsymbol{s}_*(\boldsymbol{\theta})$ unique, it is independent of ϕ_* but also can be computed very efficiently (in $\mathcal{O}(d\log d)$ time where d is the dimension) as

$$\boldsymbol{s}_*(\boldsymbol{\theta}) = \operatorname{Argmin}_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} ||\boldsymbol{y}-\boldsymbol{s}||^2. \quad (4.14)$$

This computation is the core of `GradMaPr` and is the key that makes solving (4.1), or equivalently solving (4.10), not only a possibility, but also very efficient. For the sets $\mathcal{S}^s, \mathcal{S}_l, \mathcal{S}^s_l$, the key steps of `GradMaPr` remain the same, it consists of marginalization and projection. The different instances of $\mathcal{S}\star$ only changes what set the aforementioned projection is computed on. To explain `GradMaPr` further requires an explanation of the conic structure of the sets $\mathcal{S}\star\,(\boldsymbol{\theta})$, which is what follows.

4.2.2 Representing $\mathcal{S}\star\,(\boldsymbol{\theta})$ by Linear Inequalities

Central to our efficient computation of $\boldsymbol{s}_*(\boldsymbol{\theta})$ via (4.14) are two algorithmic devices (i) Bregman's algorithm for solving linearly constrained convex optimization problems (Bregman, 1967) and (ii) The pool adjacent violators (PAV) algorithm (Best and Chakravarti, 1990). In fact the latter is a specialized invocation of the former. Both require the representation of the constraints as a set of linear inequalities, whereas the representation of $\mathcal{S}\star\,(\boldsymbol{\theta})$ described so far does not have that form. In this section we give an alternative characterizations of the sets $\mathcal{S}\star\,(\boldsymbol{\theta})$ that will enable the use of PAV and Bregman's algorithm.

Let A be the adjacent-difference matrix. Now consider the sets

$$\begin{aligned}
\mathcal{G}(\boldsymbol{\theta}) &= \{\boldsymbol{s}|A\boldsymbol{s} \leq 0\} = \mathbb{G}(\boldsymbol{\theta}),\\
\mathcal{G}_l(\boldsymbol{\theta}) &= \{\boldsymbol{s}|lA\boldsymbol{s} \leq A\boldsymbol{\theta}\} = \mathbb{G}^{\frac{1}{l}}(\boldsymbol{\theta}),\\
\mathcal{G}^s(\boldsymbol{\theta}) &= \{\boldsymbol{s}|A\boldsymbol{\theta} \leq sA\boldsymbol{s} \leq 0\} = \mathbb{G}_{\frac{1}{s}}(\boldsymbol{\theta}),\\
\mathcal{G}^s_l(\boldsymbol{\theta}) &= \{\boldsymbol{s}|lA\boldsymbol{\theta} \leq lsA\boldsymbol{s} \leq sA\boldsymbol{\theta}\} = \mathbb{G}^{\frac{1}{l}}_{\frac{1}{s}}(\boldsymbol{\theta}).
\end{aligned} \tag{4.15}$$

collectively called $\mathcal{G}\star\,(\boldsymbol{\theta})$ and $\mathbb{G}\star\,(\boldsymbol{\theta})$ respectively.

Lemma 14.

$$\begin{aligned}
\mathcal{S}(\boldsymbol{\theta}) &= \pi_{\boldsymbol{\theta}}\mathcal{G}(\boldsymbol{\theta}),\\
\mathcal{S}_l(\boldsymbol{\theta}) &= \pi_{\boldsymbol{\theta}}\mathcal{G}_l(\boldsymbol{\theta}),\\
\mathcal{S}^s(\boldsymbol{\theta}) &= \pi_{\boldsymbol{\theta}}\mathcal{G}^s(\boldsymbol{\theta}),\\
\mathcal{S}^s_l(\boldsymbol{\theta}) &= \pi_{\boldsymbol{\theta}}\mathcal{G}^s_l(\boldsymbol{\theta})
\end{aligned}$$

where $\pi_{\boldsymbol{\theta}}$ is the inverse permutation operator that sorts $\boldsymbol{\theta} = X\boldsymbol{w}$ in ascending order. When the components of $\boldsymbol{\theta}$ are not all unique, the sorting operator is also non-unique. In this case we form $\mathcal{G}\star\,(\boldsymbol{\theta})$ as described by considering the unique values of $\boldsymbol{\theta}$ only and then add equality constraint for every replicated value occurring in $\boldsymbol{\theta}$.

Proof. We show $\mathcal{S}\star(\boldsymbol{\theta}) \subset \mathcal{G}\star(\boldsymbol{\theta})$ and $\mathcal{G}\star(\boldsymbol{\theta}) \subset \mathcal{S}\star(\boldsymbol{\theta})$. The first subset relation follows from the facts that inverse gradient of $\mathcal{C}, \mathcal{C}_l, \mathcal{C}^s, \mathcal{C}^s_l$ are monotone, strongly monotone, monotone and Lipschitz continuous, and strongly monotone, Lipschitz continuous respectively. We show the second subset relation, by explicitly constructing an appropriate convex function starting from the set $\mathcal{G}\star\left(\boldsymbol{\theta}\right)$.

To see $\mathcal{G}(\boldsymbol{\theta}) \subset \mathcal{S}(\boldsymbol{\theta})$ consider the integral of the monotonic curve $(\boldsymbol{\theta}, \boldsymbol{s})$, it is clearly convex. To obtain $\mathcal{S}^s(\boldsymbol{\theta})$ from $\mathcal{G}^s(\boldsymbol{\theta})$ integrate the monotonic curve $(\boldsymbol{\theta}, \boldsymbol{s} - s\boldsymbol{\theta})$. To obtain $\mathcal{S}_l(\boldsymbol{\theta})$ from $\mathcal{G}_l(\boldsymbol{\theta})$ we use what may be called *infimal de-convolution*. Integrate $(\boldsymbol{\theta}, \boldsymbol{s})$, to form a convex function, compute its Legendre conjugate, (this will be strongly convex), subtract the function $\frac{1}{l}||\cdot||^2$, (this will be a convex function), then take its Legendre transform. □

Corollary 4. $\boldsymbol{s}_*(\boldsymbol{\theta}) = \pi_{\boldsymbol{\theta}}\left(\mathrm{Argmin}_{\boldsymbol{v}\in\mathcal{G}\star(\boldsymbol{\theta})}\, ||\boldsymbol{v} - (\pi_{\boldsymbol{\theta}})^{-1}(\boldsymbol{y})||^2\right)$

Proof. Follows from separability of $\phi(\cdot)$, theorem 7 and Lemma 14. □

Now let us get back to the central claim that $\mathrm{Argmin}_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})}\, M\star\left(\boldsymbol{s}, \boldsymbol{\theta}\right)$ and hence $\partial_{\boldsymbol{w}} \inf_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star\left(\boldsymbol{s}, \boldsymbol{\theta}\right)$ is unique and independent of ϕ_*.

Theorem 7. $\mathrm{Argmin}_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})}\, M\star(\boldsymbol{s}, \boldsymbol{\theta})$ *is unique, independent of the minimizing* ϕ_**s defined in* (4.9) *and obtained as the Euclidean projection of* $\boldsymbol{y}$ *on* $\mathcal{S}\star\left(\boldsymbol{\theta}\right)$.

Proof. From (4.11), the KKT conditions of $\min_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})}\, M\star\left(\boldsymbol{s}, \boldsymbol{\theta}\right)$ are:

$\boldsymbol{s}(\boldsymbol{\theta}) - \boldsymbol{y} \in \left([\nabla^2\phi_*]\right)^{-1}\mathcal{N}(\mathcal{S}\star\left(\boldsymbol{\theta}\right))$ and $\boldsymbol{s}(\boldsymbol{\theta}) \in \mathcal{S}\star\left(\boldsymbol{\theta}\right)$.

The matrix $\left([\nabla^2\phi_*]\right)^{-1}$ is positive definite and diagonal. Now observe that the KKT conditions are exactly the definition of the projection of $\boldsymbol{y}$ on $\mathcal{S}\star\left(\boldsymbol{\theta}\right)$ according to the squared Mahalonobis distance defined by the matrix $\left([\nabla^2\phi_*]\right)^{-1}$, which according to Corollary 4.1 is independent of $\left([\nabla^2\phi_*]\right)^{-1}$ if $\mathcal{S}\star\left(\boldsymbol{\theta}\right)$ has the conic structure of sorted vectors, as already shown in Lemma 14 and elaborated further in Section 4.2.4. Observe that the matrix $\left([\nabla^2\phi_*]\right)^{-1}$ was the only term that depended on a particular ϕ_*. This concludes the proof. □

Corollary 5. *The subgradient defined in* (4.8) *is*

$$\begin{aligned}\partial_w m(\boldsymbol{w}) &= \partial_w \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)\\ &= X^\dagger \partial_{\boldsymbol{\theta}} \inf_{\phi\in\mathcal{C}\star} D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}(X\boldsymbol{w})\big)\\ &= \underset{\phi_*\in\underset{\phi\in\mathcal{C}\star}{\mathrm{Argmin}} D_{\boldsymbol{\phi}}\left(\boldsymbol{y}\middle|\middle|(\nabla\phi)^{-1}(\boldsymbol{\theta})\right)}{\mathrm{ConvHull}} X^\dagger\{(\nabla\phi_*)^{-1}(\boldsymbol{\theta})-\boldsymbol{y}\} = X^\dagger(\boldsymbol{s}_*(\boldsymbol{\theta})-\boldsymbol{y}).\end{aligned}$$

Proof. $m(\boldsymbol{w})$ and $\inf_{\boldsymbol{s}\in\mathcal{S}\star(\boldsymbol{\theta})} M\star(\boldsymbol{s},\boldsymbol{\theta})$ are the same function. □

4.2.3 Kernelization

Observe that, on taking the regularizer $\mathfrak{R}(\boldsymbol{w})$ to be $||\boldsymbol{w}||^2$ in (4.10) we obtain

$$\boldsymbol{w}_* = \frac{1}{c}X^\dagger(\boldsymbol{s}_*(\boldsymbol{\theta})-\boldsymbol{y})$$

where the optimal $\boldsymbol{s}_*(\boldsymbol{\theta})$ has to be determined from the training set. An immediate consequence of this is that $\boldsymbol{\theta}$ and consequently the formulation can be posed entirely in terms of a kernel $K(\cdot,\cdot)$ and the parameter $\boldsymbol{\alpha} = \boldsymbol{s}_*(\boldsymbol{\theta})$ to be determined. To see this note $\boldsymbol{\theta} = X\boldsymbol{w} = XX^\dagger\boldsymbol{\alpha} = K\boldsymbol{\alpha}$ and therefore (4.1) is equivalent to

$$\min_{\boldsymbol{\alpha}} \frac{1}{n} D_\phi\big(\boldsymbol{y}\big|\big|K\boldsymbol{\alpha}\big) + c||\boldsymbol{\alpha}||_K.$$

We do not pursue this further as it lies beyond our scope, but the methods to do it is straightforward and well known (Grunwald and Dawid, 2005).

4.2.4 Convergence of `GradMaPr` in Linear Time

The gradient computation using `GradMaPr` takes finite and no more than $\mathcal{O}d\log d$ time. This comes about as a result of reducing the gradient computation to variants of isotonic regression which we then solve in time upper bounded by linear function of the dimension.

To achieve this we will use the fact that the pool adjacent violators (PAV) algorithm can compute the squared Euclidean projection on the monotone cone in linear time. This by itself is not sufficient, but for two other results we have shown (i) the Mahalonobis projection on the same cone defined by any diagonal matrix coincides with the squared Euclidean projection, and (ii) even if the monotone-conic structure is not apparent, it can, in our case, be obtained following some affine transformations, detailed further in this section. After the said transformations have been applied, the constraint set still may not be conic, for example, when we have Lipschitz constraints on $(\nabla\phi(\cdot))^{-1}$. In such cases, however, the constraint set will be of the form of an monotone cone intersected with an affine manifold (linear equality constraint) of special structure. For this special structure, we shall show that PAV followed by a single update of Bregman's projection obtains the solution regardless of the diagonal matrix used to define the Mahalonobis projection.

That the PAV algorithm can compute isotonic regression in linear time is known. However, it appears that algorithm employed to solve the Lipschitz continuity constrained variant, and the consequential improved time complexity bound achieved, is new. It improves upon the best known bound for solving isotonic regression under Lipschitz continuity constraints. Indeed the journal paper (Yeganova and Wilbur, 2009) is exclusively on developing a finite time, quadratic time complexity algorithm for the problem, whereas here it is solved in finite time but with linear complexity and is further invariant to changes in the the diagonal matrix used to define the Mahalonobis projection.

Recall from Theorem 7 that $\boldsymbol{s}_*(\boldsymbol{\theta})$ is the projection of $\boldsymbol{y}$ on the set $\mathcal{S}\star\,(\boldsymbol{\theta})$, and Lemma 14 provides a characterization of $\mathcal{S}\star\,(\boldsymbol{\theta})$ in terms of linear inequalities. Clearly Bregman's algorithm applies. Rather than invoking Bregman's algorithm generically, we exploit the special structure present in $\mathcal{S}\star(\boldsymbol{\theta})$, in particular the fact that the equivalent linear inequalities are in terms of the adjacent-difference operator A. This form is particularly suited to the pool adjacent violators algorithm (PAV).

Pool Adjacent Violators

The pool adjacent violators algorithm solves the following problem

$$\min_{\boldsymbol{v}} ||\boldsymbol{v} - \boldsymbol{y}||^2 \quad \text{s.t.} \quad A\boldsymbol{v} \leqq \mathbf{0} \tag{4.16}$$

called the isotonic regression. A is the adjacent difference matrix and the symbol $\leqq$ indicates that each row of $A\boldsymbol{v} \leqq \mathbf{0}$ may either be an equality constraint or an inequality constraint.

PAV is essentially an instance of Bregman's algorithm using block projections. It runs in *finite time* and a straight-forward implementation scales as $\mathcal{O}(d^2)$ where d is the dimensions. However Grotzinger and Witzgall (Grotzinger and Witzgall, 1984) observed that if implemented carefully it remarkably has linear complexity. It can, however, be easily adapted to handle both lower and upper bound constraints on the components of $\boldsymbol{v}$ as well as equality constraints on some of its adjacent components, all while maintaining the same time complexity.

In the remaining of the section we adapt the PAV algorithm to the different constraint sets $\mathcal{G}(\boldsymbol{\theta})$, $\mathcal{G}_l(\boldsymbol{\theta})$, $\mathcal{G}^s(\boldsymbol{\theta})$ and $\mathcal{G}^s_l(\boldsymbol{\theta})$. The key is to ensure linear runtime of the algorithm.

Restricted Output Space: As mentioned earlier in the introduction, one may have additional information about the structure of the output space of each scalar valued prediction y_i. It arises for example when predicting probabilities, in that case we know that $(\nabla\phi)^{-1}(\langle \boldsymbol{x}_i, \boldsymbol{w}\rangle) \in [0, 1]\, \forall i$. Since we are restricted to convex output spaces and hence intervals, such structure can be easily incorporated by the addition of lower and upper bound inequalities to our characterization of the sets $\mathcal{G}(\boldsymbol{\theta})$, $\mathcal{G}_l(\boldsymbol{\theta})$, $\mathcal{G}^s(\boldsymbol{\theta})$ and $\mathcal{G}^s_l(\boldsymbol{\theta})$. For the pav algorithm, this causes no loss in computational complexity. These additional lower and upper bound inequalities are dropped from our description of sets $\mathcal{G}(\boldsymbol{\theta})$, $\mathcal{G}_l(\boldsymbol{\theta})$, $\mathcal{G}^s(\boldsymbol{\theta})$ and $\mathcal{G}^s_l(\boldsymbol{\theta})$ for notational simplicity. Note however that if the training $\boldsymbol{y}$ itself is constrained to be in

the Cartesian product of such intervals, no extra inequalities need be added as the training prediction is by nature of the pav algorithm constrained to lie in the interval spanned by $\boldsymbol{y}$.

Case $\mathcal{G}(\boldsymbol{\theta})$:

Here the constraint set used by the PAV algorithm, namely $A(\boldsymbol{v}) \leq \mathbf{0}$ coincides with $\mathcal{G}(\boldsymbol{\theta})$ therefore PAV can be used directly with no change.

Case $\mathcal{G}_l(\boldsymbol{\theta})$:

In this case the constraint set is given by $lA\boldsymbol{s} \leq A\boldsymbol{\theta}$ and thus it does not exactly match the form used by the pav problem. However with the simple change of variable variables $\tilde{\boldsymbol{s}}(\boldsymbol{\theta}) = (l\boldsymbol{s} - \boldsymbol{\theta})$ the pav formulation is recovered exactly as

$$\tilde{\boldsymbol{s}}(\boldsymbol{\theta}) = \pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \left(\mathrm{Argmin}_{\tilde{\boldsymbol{s}}} ||\tilde{\boldsymbol{s}} - (\pi_{\boldsymbol{\theta}})^{-1}(\boldsymbol{y})||^2 \quad \text{s.t.} \quad A\tilde{\boldsymbol{s}} \leq \mathbf{0} \right).$$

Cases $\mathcal{G}_l(\boldsymbol{\theta}), \mathcal{G}_l^s(\boldsymbol{\theta})$:

Here unlike the two previous cases the inequalities are constrained both from above and below:

$$A\boldsymbol{s} \geq \frac{1}{s} A\boldsymbol{\theta} \quad \text{and} \quad A\boldsymbol{s} \leq \frac{1}{l} A\boldsymbol{\theta}.$$

Since we can recover $\mathcal{G}_l(\boldsymbol{\theta})$ as a special case of $\mathcal{G}_l^s(\boldsymbol{\theta})$ we discuss the latter only.

To our knowledge the algorithm with the best runtime complexity for solving the isotonic regression problem over the set $\mathcal{G}_l(\boldsymbol{\theta})$ is the Lipschitz PAV algorithm (Yeganova and Wilbur, 2009) that has a finite time complexity of $O(d^2)$ where d is the dimensionality. Here we obtain an order $\mathcal{O}(d)$ improvement by proposing an alternative algorithm that has a finite time complexity bounded by $\mathcal{O}(d)$ in the dimension. To explain the algorithm let us split the variable $\boldsymbol{s}$ (and the corresponding inequalities) to obtain $A\boldsymbol{s}_+ \leq -\frac{1}{s} A\boldsymbol{\theta}, \quad A\boldsymbol{s}_- \leq \frac{1}{l} A\boldsymbol{\theta}, \quad \text{and} \quad 0 = \boldsymbol{s}_+ + \boldsymbol{s}_-$.

We write the constraints in a more suggestive form by concatenating the variables

as follows: $\binom{\boldsymbol{s}_+}{\boldsymbol{s}_-}$.

$$\begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & A \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_+ \\ \boldsymbol{s}_- \end{pmatrix} \leq \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & A \end{pmatrix} \begin{pmatrix} -\frac{\boldsymbol{\theta}}{s} \\ \frac{\boldsymbol{\theta}}{l} \end{pmatrix} \tag{4.17}$$

$$\begin{pmatrix} I & I \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_+ \\ \boldsymbol{s}_- \end{pmatrix} = 0 \tag{4.18}$$

This variable splitting induces an equivalent/conformal split on $\boldsymbol{y}$ as $\boldsymbol{y}_-$, $\boldsymbol{y}_+$ and in the cost function as follows.

$$\min_{\binom{\boldsymbol{s}_+}{\boldsymbol{s}_-}} \left\| \begin{pmatrix} \boldsymbol{s}_+ \\ \boldsymbol{s}_- \end{pmatrix} - (\pi_{\boldsymbol{\theta}})^{-1} \begin{pmatrix} -\boldsymbol{y}_+ \\ \boldsymbol{y}_- \end{pmatrix} \right\|^2 \tag{4.19}$$

Now we can apply Bregman's algorithm to the cost function (4.19) subject to the constraints (4.17) and (4.18). Note that the variables $\boldsymbol{s}_+$ and $\boldsymbol{s}_-$ are decoupled in the constraints (4.17), as well as in the cost function, hence the Bregman updates can be computed in parallel using PAV in linear time (see section (4.2.4)). In the next step we need to project the solution obtained on the constraint (4.18) leading to the update (see section (2.2.3))

$$\begin{pmatrix} \boldsymbol{s}_+ \\ \boldsymbol{s}_- \end{pmatrix}^{t+1} = \begin{pmatrix} \boldsymbol{s}_+ \\ \boldsymbol{s}_- \end{pmatrix}^{t} + c \begin{pmatrix} I \\ I \end{pmatrix}. \tag{4.20}$$

However, since this update does not violate the constraints (4.17) this terminates the iterations of Bregman's algorithm and we obtain the optimum.

K **Invariance** We have established before that the minimizing the second argument of a Bregman divergence over the monotone cone is independent of the Bregman divergence as long as it is separable. As a result Mahalonobis distance projections on the monotone cone is invariant as long as it is defined by K, a diagonal positive definite matrix. Does the invariance also hold for this $\mathcal{G}_l(\boldsymbol{\theta}), \mathcal{G}_l^s(\boldsymbol{\theta})$ case ?

Note that equation (4.17) defines projections on monotone cones, so they are clearly unaffected by K. What remains to be shown is that constraint (4.20) remains unaffected as well. Observe that because of variable splitting into $\boldsymbol{y}_-$, $\boldsymbol{y}_+$, the matrix K gets replicated along the diagonal in (4.19), therefore (4.20) continues to maintain the constraint (4.18).

Convergence Rates of Realizable Algorithms: Now that we can compute the gradient of $m(w)$ we can realize the algorithms described in Table 4.2. If we optimize over $\phi \in \mathcal{C}^s$ (equivalently over $\boldsymbol{s}(\boldsymbol{\theta}) \in \mathcal{S}_l(\boldsymbol{\theta})$ with $l = 1/s$) Lemma 13 ensures that the gradient will have a Lipschitz constant L, this coupled with accelerated gradient descent obtains a convergence rate of $\mathcal{O}(\frac{1}{T^2})$. Optimizing over $\mathcal{C}$ (equivalently over $\mathcal{S}$) obtains convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$. Both the rates are optimal for first order methods uniformly in the dimension.

4.3 Prediction

We consider two types of prediction problems:

1. predicting the y corresponding to an unseen test point $\boldsymbol{x}$ and
2. predicting the complete order over the set of new test items represented as rows of an unseen test matrix X_t.

Recall that the prediction is given by $(\nabla)^{-1}\phi(\langle \boldsymbol{x}, \boldsymbol{w} \rangle)$. Although we obtain $\boldsymbol{w}$ explicitly at the end of the training phase, an explicit representation of $\phi(\cdot)$ is not obtained. In fact $\phi(\cdot)$ cannot be obtained uniquely because the cost function is only convex in $\phi(\cdot)$ and not strictly convex. We only know the optimal $\phi(\cdot)$ via its inverse gradients at the training points. For a new test point $\boldsymbol{x}$ we can, however, narrow the prediction down to an interval.

Let $\boldsymbol{w}_*$ be the optimal $\boldsymbol{w}$ returned by the algorithm, let $\theta_t = \langle \boldsymbol{w}_*, \boldsymbol{x} \rangle$, $\theta_l = \max_{\langle X(i), \boldsymbol{w} \rangle \leq \theta_t} \langle X(i), \boldsymbol{w} \rangle$, $\theta_u = \min_{\langle X(i), \boldsymbol{w} \rangle \geq \theta_t} \langle X(i), \boldsymbol{w} \rangle$ and the corresponding y's be y_l, y_u. Then the prediction y corresponding to $\boldsymbol{x}$ is given as:

$$y \in \begin{cases} [y_l, y_u] & \text{when (4.1) optimized over } \mathcal{C} \\ [\max(y_l, y_u - L(\theta_u - \theta)), \min(y_u, y_l + s(\theta - \theta_l))] & \text{when (4.1) optimized over } \mathcal{C}^s \\ [\max(y_l, y_u - l(\theta_u - \theta), \min(y_u, y_l + l(\theta - \theta_l))] & \text{when (4.1) optimized over } \mathcal{C}_l. \end{cases}$$

Continuity: Note that the prediction function is a point-to-set mapping, note in particular that this point-to-set-mapping is continuous at the training points, where continuity of a point-to-set-map is defined in the usual way (Rockafellar, 1996) as follows: A point-to-set-map $y(\boldsymbol{x})$ is continuous if for all sequences $\boldsymbol{x}_t \to \boldsymbol{x}$ there exists a $y_t \to y$ such that $y_t \in y(\boldsymbol{x}_t)$.

Recovering $\phi(\cdot)$: Although we cannot recover an unique $\phi(\cdot)$ one instance it can be recovered upto agreement with the training data. To obtain such an estimate, one needs to select a continuous function from the point-to-set mapping $\boldsymbol{x} \mapsto \bar{y}(\boldsymbol{x})$, where we use $\bar{y}$ to indicate a selection. Taking the Legendre dual of the integral of the curve $\boldsymbol{x} \mapsto \bar{y}(\boldsymbol{x})$ obtains a desired $\phi(\cdot)$.

Restricted Output Space: If we have incorporated the restriction on the outputs space in the definition of $\mathcal{G}\star(\boldsymbol{\theta})$ as indicated in Section 4.2.4, there is little that needs to be done at prediction time. If test $\boldsymbol{x}$ is such that $\langle \boldsymbol{w}_*, \boldsymbol{x} \rangle \in [\min_i \langle \boldsymbol{w}_*, \boldsymbol{x}_i \rangle, \max_i \langle \boldsymbol{w}_*, \boldsymbol{x}_i \rangle]$ nothing needs to be done as the prediction function $y(\boldsymbol{x})$ will automatically guarantee the output space interval constraints. On the other hand if $\langle \boldsymbol{w}_*, \boldsymbol{x} \rangle$ lies outside of the range thresholding may be necessary.

4.4 Non-agnostic Case

As a pedagogic shortcut we have motivated the cost function (4.1) using the notion of a vector $\boldsymbol{u}$ that achieves $\boldsymbol{y} = g(X\boldsymbol{w}) = (\nabla\phi)^{-1}(X\boldsymbol{w})$ exactly. The optimization algorithms presented, however, do not require the existence of such a $\boldsymbol{u}$. They obtain the minimum

regardless. If, however, there is prior knowledge to indicate that such a *perfect* $\boldsymbol{u}$ exists, much more efficient techniques may be applied to recover it.

First observe that the *perfect* $\boldsymbol{u}$ assumption implies the following $\{\exists \phi \in \mathcal{C}\star \text{ s.t. } X\boldsymbol{u} \in \nabla\phi(\boldsymbol{y})\} \equiv \{X\boldsymbol{u} \in (\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star\,(\pi_{\boldsymbol{y}}(\boldsymbol{y}))\}$. When the regularization on $\boldsymbol{w}$ is specified using a set $\mathcal{W}$ the vector $\boldsymbol{u}$ can be obtained as the following convex feasibility problem

$$\boldsymbol{\theta} \in \{(\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star\,(\pi_{\boldsymbol{y}}(\boldsymbol{y})\} \cap \{X\mathcal{W}\}. \tag{4.21}$$

Any such convex feasibility problem may be solved by both the sequential as well as the parallel Bregman's algorithm (see section 2.2.3), the specific Bregman divergence used in Bregman's algorithm to obtain convex feasibility, does not matter. It is therefore advantageous to choose the divergence for which the projections are the simplest to compute. The Bregman projection on the set $\{(\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star\,(\pi_{\boldsymbol{y}}(\boldsymbol{y})\}$ can be computed in linear time by the PAV variants discussed in Section 4.2.4, as long as the Bregman divergence is separable.

In general, computationally convenient projections on two different sets may be obtained by two different Bregman divergences. Using different Bregman divergences, tailored to the different sets is well explored in the context of these problems called the split feasibility problem (Censor and Elfving, 1994).

For our framework, two cases are particularly convenient: (i) $\mathcal{W}$ is an ℓ_2 ball and (ii) $\mathcal{W} = \{\boldsymbol{z}|\ \ ||\boldsymbol{z}||^2_{X^\dagger X} \leq L\}$. Choosing the Bregman divergence to be squared Euclidean, we obtain the projection on $\{(\pi_{\boldsymbol{y}})^{-1}\mathbb{G}\star(\pi_{\boldsymbol{y}}(\boldsymbol{y})\}$ in linear time by the PAV algorithm and the projection on the set $\mathcal{W}$ reduces to a regularized least squares in case of (i) and is obtained in closed form for case (ii). Both the solutions can be obtained in time linear in the dimension. In this case we obtain an overall linear convergence rate (Deutsch and Hundal, 2006), as is the case if we apply ADMM to the same problem (Luo, 2012).

It is known that if the intersection of the sets specified in the CFP problem is non-empty both the sequential and parallel Bregman's algorithm converges to a feasible point (Censor and Zenios, 1997). On the other hand if the intersection is empty, the parallel Breg-

man's algorithm converges to a point that minimizes the sum of the Bregman divergences from the specified sets. On the other hand the sequential Bregman algorithm converges to a limit cycle, where the projections on each of the sets themselves converge (and thus exhibits cyclic behavior).

Note that the case where the intersection is empty conforms to the agnostic case, i.e. there is no $\boldsymbol{u}$ that achieves a 0 loss $D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big)$. It is important to remember, however, that though the parallel Bregman's algorithm obtains a solution in this agnostic case it does not optimize the cost function (4.1) and is dependent on the $\phi(\cdot)$ used to obtain the Bregman projections.

4.5 Sensitivity to Perturbation

So far we have largely motivated our cost function (4.1) assuming that $\boldsymbol{y}$ equals $g(X\boldsymbol{u})$ *exactly*. An equivalent re-statement of this unrealistic assumption, is that we obtain a perfect empirical estimate of the conditional expectation (from an unknown GLM). This was a pedagogic device, used only to motivate the cost function. The proposed algorithm minimizes the cost function regardless of whether the noise-free assumption holds or not.

In practice we only have access to samples drawn from the conditional distribution. Thus a vital question is: how well does the proposed algorithm perform in a more realistic setting. We denote our estimates by $\boldsymbol{w}_*$ and $\tilde{\boldsymbol{w}}_*$, they correspond to $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ respectively and hence satisfy the following conditions:

$$\boldsymbol{w}_* = \mathrm{Argmin}_{\phi\in\mathcal{C}_\star,\boldsymbol{w}}\, D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\big|\big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big) + c\mathfrak{R}(\boldsymbol{w}) \tag{4.22}$$

$$\tilde{\boldsymbol{w}}_* = \mathrm{Argmin}_{\phi\in\mathcal{C}_\star,\boldsymbol{w}}\, D_{\boldsymbol{\phi}}\big(\tilde{\boldsymbol{y}}\big|\big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big) + c\mathfrak{R}(\boldsymbol{w}). \tag{4.23}$$

Now we quantify

- how far can the estimate $\tilde{\boldsymbol{w}}_*$ be from $\boldsymbol{w}_*$ when the $\tilde{\boldsymbol{y}}$ used by the algorithm is $||\tilde{\boldsymbol{y}} - g(X\boldsymbol{u})||$ away from $g(X\boldsymbol{u})$, and

- with what probability does the proposed algorithm recover $\boldsymbol{w}_*$ with accuracy $||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*|| \leq \epsilon$.

The latter is computed without assuming a *particular* form of the exponential family that generated the sample $\tilde{y}_i$s, but with the assumption, that $\tilde{y}_i$ were drawn independently conditioned on $\boldsymbol{x}_i$ from some unknown exponential family satisfying some curvature assumptions on its (negative)-entropy, for example: the negative entropy is s-strongly convex, or δ uniformly convex. Both are proven by quite elementary techniques.

4.5.1 Deterministic Case

Lemma 15. *(Rockafellar, 1996) Let $\boldsymbol{q}_i, \boldsymbol{q}_j$ be the squared Euclidean projections of $\boldsymbol{p}_i, \boldsymbol{p}_j$ on any closed convex set $\mathcal{C}$, i.e. $\boldsymbol{q}_i = \mathrm{Proj}_{\mathcal{C}}(\boldsymbol{p}_i) = \mathrm{Argmin}_{\boldsymbol{x}\in\mathcal{C}} ||\boldsymbol{x} - \boldsymbol{p}_i||^2$. Then $||\boldsymbol{q}_i - \boldsymbol{q}_j|| \leq ||\boldsymbol{p}_i - \boldsymbol{p}_j||$.*

Lemma 16. *Let $\boldsymbol{x}_* = \mathrm{Argmin}_{\boldsymbol{y}} f(\boldsymbol{y})$ where $f(\cdot)$ is a differentiable, $s(K)$-strongly convex function under the $||\cdot||_K$ norm then $||\boldsymbol{x}_* - \boldsymbol{x}||_K \leq ||\nabla f(\boldsymbol{x})||_{K^{-1}}$.*

Proof. The gradient of a $s(K)$-strongly convex function is $s(K)$-strongly monotone, therefore $\langle \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}_*), \boldsymbol{x} - \boldsymbol{x}_* \rangle \geq s(K)||\boldsymbol{x} - \boldsymbol{x}_*||_K^2$. Invoking Holder's inequality with the dual norms $||\cdot||_K$ and $||\cdot||_{K^{-1}}$ we obtain $||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}_*)||_{K^{-1}}\ ||\boldsymbol{x} - \boldsymbol{x}_*||_K > s(K)||\boldsymbol{x} - \boldsymbol{x}_*||_K^2$. □

Lemma 17. *Let $\mathbb{R}^n \ni \boldsymbol{y} = g(X\boldsymbol{u})$ with $g \in \{(\nabla\phi)^{-1} | \phi \in \mathcal{C}\star\}$. If expression* (4.22) *is minimized over ϕ in the class $\mathcal{C}_l^s$ then $\|\boldsymbol{u} - \boldsymbol{w}_*\|_{A^\dagger A} \leq 2l\sqrt{2\frac{\mathfrak{R}(\boldsymbol{u})}{s}}$ and if expression* (4.22) *is minimized with ϕ in the class of uniformly convex function with modulus of uniform convexity $\delta(\cdot)$ and with $L-$Lipschitz continuous gradient then $\boldsymbol{w}_*\|_{A^\dagger A} \leq 2l\sqrt{\delta^{-1}(2\mathfrak{R}(\boldsymbol{u}))}$*

Theorem 8. *Let $\mathbb{R}^n \ni \boldsymbol{y} = g(X\boldsymbol{u})$ with $g \in \{(\nabla\phi)^{-1} | \phi \in \mathcal{C}\star\}$ and*

$$\tilde{\boldsymbol{w}}_* = \underset{\phi\in\mathcal{C}\star, \boldsymbol{w}}{\mathrm{Argmin}} D_{\boldsymbol{\phi}}\big(\tilde{\boldsymbol{y}} \Big|\Big| (\nabla\phi)^{-1}(X\boldsymbol{w})\big) + c\mathfrak{R}(\boldsymbol{w}).$$

Let the regularizer $\mathfrak{R}(\cdot)$ be continuously differentiable and $s_{\mathfrak{R}}(K)$-strongly convex where K is any positive diagonal matrix.

$$||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*||_K \leq \frac{||\tilde{\boldsymbol{y}} - \boldsymbol{y}||_{XK^{-1}X^\dagger}}{cs_{\mathfrak{R}}(K)}. \tag{4.24}$$

Proof. $\boldsymbol{w}_*$ is the stationary point of $\min_{\boldsymbol{s}\in\mathcal{S}\star} \frac{1}{n}M(\boldsymbol{s},\boldsymbol{w})+\mathfrak{R}(\boldsymbol{w}) = \frac{1}{n}m(\boldsymbol{w})+\mathfrak{R}(\boldsymbol{w})$. From (4.12) we have $\nabla_{\boldsymbol{w}} m(\boldsymbol{w}) = X^\dagger \operatorname{Proj}_{\mathcal{S}\star}(\boldsymbol{y})$. When $\boldsymbol{y}$ is corrupted into $\tilde{\boldsymbol{y}}$ we obtain the corrupted gradient $\nabla_{\boldsymbol{w}}\tilde{m}(\boldsymbol{w})$ as $X^\dagger \operatorname{Proj}_{\mathcal{S}\star}(\tilde{\boldsymbol{y}})$. Let $\tilde{\boldsymbol{w}}_*$ be the stationary point of $\tilde{m}(\boldsymbol{w}) + \mathfrak{R}(\boldsymbol{w})$.

$$\begin{aligned}
||\tilde{\boldsymbol{y}} - \boldsymbol{y}||_{XK^{-1}X^\dagger} &\geq ||\nabla_{\boldsymbol{\theta}}\tilde{m}(\boldsymbol{w}_*) - \nabla_{\boldsymbol{\theta}} m(\boldsymbol{w}_*)||_{K^{-1}} \\
&= ||\nabla_{\boldsymbol{\theta}}\tilde{m}(\boldsymbol{w}_*)||_{K^{-1}} \\
&\geq cs_{\mathfrak{R}}(K)||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*||_K.
\end{aligned}$$

□

K **Invariance:** A distinguishing characteristic of the bound (4.24) is that one can tighten them by selecting K. We emphasize that the algorithm itself is oblivious to the choice of K, it is the bound that holds for all K that are positive definite and diagonal, allowing it to be tightened. The reason it is possible to do so is because of the property that the projection on $\mathcal{S}\star$ is invariant to the choice of K.

4.5.2 Probabilistic Case

So far in this section we have not made any probabilistic assumption on how $\tilde{y}_i$ is generated. Now we shall assume that $g(\cdot)$ is the expectation function of a canonical GLM (McCulloch and Searle, 2001), equivalently:

$$P(y|\boldsymbol{x}) = e^{\langle \boldsymbol{x},\boldsymbol{u}\rangle y - \phi^*(\langle \boldsymbol{x},\boldsymbol{u}\rangle)}.$$

If each component of $\tilde{\boldsymbol{y}}$ is a conditionally independent sample drawn from the distribution above, what can one claim about the probability $P(||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*||_K \leq t)$. We bound this probability simply be recognizing that this can be bounded as

$$P(||\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*||_K \leq t) \geq P\left(||\tilde{\boldsymbol{y}} - \boldsymbol{y}||_{XK^{-1}X^\dagger} \leq cs_{\mathfrak{R}}(K)t\right). \tag{4.25}$$

We will first provide a bound assuming $\mathcal{C}\star = \mathcal{C}^s$ i.e the set of all s-strongly convex functions, and then relax the restriction to the larger class of *uniformly convex functions* with a known modulus of convexity. Both of them are specializations of Cramer's theorem.

Theorem 9. *Let $\boldsymbol{y}$ have the probability density $P(y|\boldsymbol{x}) = e^{\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle - \phi^*(\boldsymbol{\theta})}$ and let the entropy function be $s(XK^{-1}X^\dagger)$-strongly convex. Then*

$$P(\|\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*\|_K) \geq 1 - \exp\left(-\frac{\sigma}{2}ts(K)^2\right)$$

for $s(K)$ the modulus of strong convexity of the regularizer we use

Proof. Plugging in the result that a $s-$strongly convex function is uniformly convex with modulus $\delta(||\boldsymbol{x}||) = s||\boldsymbol{x}||^2$ in Theorem 16 in Appendix B.1 obtains the result. □

Theorem 10. *Let $\boldsymbol{y}$ have the probability density $P(y|\boldsymbol{x}) = e^{\langle \boldsymbol{\theta}, \boldsymbol{y}\rangle - \phi^*(\boldsymbol{\theta})}$ and let the entropy function be uniformly convex with the modulus function $\delta(\cdot)$ with norm $||\cdot||_{K^{-1}}$. Then*

$$P(\|\tilde{\boldsymbol{w}}_* - \boldsymbol{w}_*\|_K) \geq 1 - \exp\left(-\delta(ts(K))\right)$$

Proof. Follows directly from Theorem 16 in Appendix B.1. □

4.6 Comparing with Isotron

Kalai and Sastri introduced `isotron` (Kalai and Sastry, 2009) updates for which they showed performance guarantees for the loss $||\boldsymbol{y} - g(X\boldsymbol{w})||^2$ for an unknown but Lipschitz

continuous monotone function $g(\cdot)$, in contrast our focus has been on keeping our estimate $\tilde{w_*}$ close to $\boldsymbol{u}$. The first surprising and impressive fact about `isotron` is that its non-convex cost function admits such guarantees, especially when the updated parameters were not shown to converge either to the local or to the global optimum of the cost function, or to anything at all. The `isotron` update was more stated than derived, adding to the mystery. This naturally provokes the question, where do the updates come from, or stated differently, can those updates be derived by following some standard optimization methodology.

Comparing the Isotron update and its improved variant `glmtron` , with the ones proposed here lifts the mystery. One can see that the `isotron` update is upto differences in learning rate, the same as the gradient descent update derived for the $\mathcal{G}(\boldsymbol{\theta})$ case, whereas the `glmtron` update is upto differences in learning rate, the same as the gradient descent update derived for the $\mathcal{G}\star\ (\boldsymbol{\theta})$ case. Both `isotron` and `glmtron` use, what in our framework would be updates with learning rate fixed at unity.

Thus `isotron` and `glmtron` updates are actually unit step size gradient descent updates on the cost function (4.1) rather than of $||\boldsymbol{y} - g(X\boldsymbol{w})||^2$. As much as this observation sheds new light on `isotron` and `glmtron` , it also exposes one of their rectifiable limitations, that is, using step size fixed at unity. As shown in Section 4.2, (Table 4.2 right) considerable acceleration may be obtained by exploiting the smoothness properties of the gradients, especially so for `glmtron` , because its gradients inherit the Lipschitz smoothness from the cost function (4.1).

To answer why should one even consider minimizing (4.1) when one is concerned with the loss $||\boldsymbol{y} - g(X\boldsymbol{w})||^2$, one only needs to realize that under the Lipschitz continuity assumption they make on $g(\cdot)$ (equivalent to strong convexity assumptions on $\phi(\cdot)$), formulation (4.1) is a convex upper bound of $||\boldsymbol{y} - g(X\boldsymbol{w})||^2$ (assuming the same regularization).

$$D_{\boldsymbol{\phi}}\big(\boldsymbol{y}\Big|\Big|(\nabla\phi)^{-1}\left(X\boldsymbol{w}\right)\big) \geq \frac{s}{2}||\boldsymbol{y} - g(X\boldsymbol{w})||^2$$

Thus in addition to being interesting in its own right formulation (4.1) also turns out to

be an effective surrogate function (Reid and Williamson, 2009) for the nonconvex loss $||\boldsymbol{y} - g(X\boldsymbol{w})||^2$.

Comparison of the Results:

In this section we compare the nature of the results obtained here with those obtained in (Kalai and Sastry, 2009) and its improved variant (Kakade et al., 2011). First we note that the current work was not motivated by the need to provide a surrogate function view of the `isotron` and `glmtron` algorithms. Given the independent development the connection came as a pleasant surprise. In spite of the similarities, there are some significant differences in the results shown. We believe quite a few can be carried over to the other.

1. For the non-realizable case `isotron` and `glmtron` analysis applies to arbitrary densities whose conditional expectation operator is Lipschitz continuous and monotone. The corresponding analysis here considers a wider class of expectation operators (those that are Holder continuous) it is less general in that it only considers exponential family densities satisfying those constraints.

   To be comparable in generality with `isotron` and `glmtron` it needs to be shown that exponential family densities satisfying those constraints form a dense cover of arbitrary densities whose conditional expectation operator is Lipschitz continuous and monotone. Given that maximum entropy under constraints also obtains the density that is mini-max distant in the KL sense of all densities that satisfy the same constraints (Grunwald and Dawid, 2004), we are hopeful that the dense cover condition holds.

2. `isotron` and `glmtron` algorithms and the associated analysis apply only to the Lipschitz continuous case, whereas those developed here apply to a larger class of Holder continuous transform $g(\cdot)$.

3. For the realizable case, i.e. when a $\boldsymbol{u}$ exists such that $\boldsymbol{y} = g(X\boldsymbol{u})$, `isotron` anal-

ysis obtains a convergence rate of $\mathcal{O}(\frac{1}{T})$ whereas for the same realizable case the projection methods discussed in Section 4.4 obtain exponential (also called linear) convergence, i.e. $\mathcal{O}(\exp -cT)$. Furthermore unlike the `isotron` analysis Lipschitz continuity is not required.

4. Since the `isotron` and `glmtron` analysis is for vectors $\boldsymbol{x}$ that satisfy $||\boldsymbol{x}|| \leq 1$ it hides the nature of dependence of the convergence rates on the size of the input. This is particularly relevant to bounds obtained in Section 4.5 because they allow a choice over K to mitigate to a large extent the effects of a badly conditioned input. Often there is predictive signal in the size of the input and although normalization on one hand will make the `isotron` and `glmtron` analysis applicable, it will also erase predictive information if present.

5. The non-realizable case is also analyzed in the `isotron` and `glmtron` papers (Kalai and Sastry, 2009), (Kakade et al., 2011). The practicability of the corresponding `isotron` algorithm is, as admitted by its authors, significantly weakened because it requires m sets of T examples with $m > \mathcal{O}(T \log(T)/l)^2$ (l is the Lipschitz constant assumed on $g(\cdot)$) to provide $\mathcal{O}(\frac{1}{T})$ bound on the expected error. This is significantly salvaged in `glmtron` but results are not comparable with ours.

4.7 Revisiting the Cost Function

We would like to highlight what (4.1) accomplishes in terms of maximum likelihood. It might be tempting to interpret it as if we are choosing a particular member over all exponential family distributions that maximizes the likelihood of the observed data. This is not what (4.1) optimizes. A careful study of the series of equalities show that though minimizing the Bregman divergence is indeed equivalent to maximum likelihood when $\phi(\cdot)$ is fixed, that interpretation does not hold when one optimise's $\phi(\cdot)$ because the term $\log P(\boldsymbol{y} \mid \boldsymbol{\theta}^*)$ is no longer constant.

We provide the following (equivalent) interpretations that can serve as alternative formulation statements

- $(\nabla\phi)^{-1}(\cdot)$ is the expectation function of the exponential family density with negative entropy $\phi(\cdot)$, thus the cost function clearly tries to match the empirical expectation over the true expectation over the family $\mathcal{G}(L)$ by minimizing the Bregman loss induced on the expectation parameter space.

- Consider a measurable space $\mathcal{Y}$ (with different measures defined on it) and the set $\mathcal{M}(s)$ of all exponential family densities with expectation s and a $\frac{1}{l}$ strongly concave entropy function. $\mathcal{Y}^*$, the dual of $\mathcal{Y}$ is the space of all linear functions defined on $\mathcal{Y}$ and serves as the container of the parameter space of $\mathcal{M}(s)$. Consider the set $\mathcal{M}(\theta)$ of all exponential family densities over $\mathcal{Y}$ whose natural parameter space intersects $\{\theta = \langle \boldsymbol{x}, \boldsymbol{w} \rangle \,|\boldsymbol{w} \in \mathcal{W}, \boldsymbol{x} \in \mathcal{X}\}$. Formulation (4.1) minimizes the KL divergence $\mathrm{KL}\left(\mathcal{M}(s) \| \mathcal{M}(\theta)\right)$. KL divergence is not defined unless the measures are absolutely continuous, this further restricts the optimization to that subset of $\mathcal{M}(\theta)$ that has the same log partition function as the dual of the negative entropy function.

- Again consider the set of densities $\mathcal{M}(s)$. Each member will be associated with a corresponding natural parameter space Θ. The formulation minimizes the "distance" between this natural parameter space and $\{\theta = \langle \boldsymbol{x}, \boldsymbol{w} \rangle \,|\boldsymbol{w} \in \mathcal{W}, \boldsymbol{x} \in \mathcal{X}\}$ measured according to the Bregman divergence induced on the natural parameter space Θ by the log partition function that is dual to the negative entropy.

Chapter 5

Consensus Ranking Using Bregman Divergences

The key task addressed in this chapter is that of consensus-based unsupervised ranking of vertices of a graph. It turns out that pagerank is a special case of our proposed method where consensus is required only at an inter-vertex level, in a way that will be elaborated further. We begin with a motivating example:

Alice, Bob and Carol are participating in a small academic conference where each person is allowed to submit only one single-author paper and each author must review all submissions. The rules of this hypothetical conference have been engineered for pedagogical purposes. Alice is a well recognized expert, so it is desired that her reviews count for more. However, rather than recognizing her "expertise" as a self declared quantity, a measure of her level of expertise is designed to emerge through a process of social consensus. This process is modeled at two levels: "local" and "global", or equivalently "intra-vertex"" and "inter-vertex" respectively. Defining this consensus algorithmically is the subject of this chapter.

We assume that reviewers evaluate the papers on the basis of multiple criteria. Each reviewer is allowed to have a set of personal criteria according to which they assign a nor-

malized score to a paper. It is possible that Alice, Bob and Carol have different notions about what constitutes a good paper. Bob's overall score for Carol's paper is obtained as a weighted average of Bob's multi-criteria scores for Carol's paper. Bob's criteria may have little or no overlap with other's, however, the weights assigned to his criteria are decided through a process of "local" consensus. Other reviewers have influence on the weights attached to each of Bob's criteria. One may ask why not weigh the personal criteria uniformly. Non-uniform weights are used to account for situations were Bob includes a criterion that others might not deem very important, for example "how many of my own papers did the reviewed paper refer to".

The consensually agreed upon weights on Bob's criteria only define the scores given by Bob. To obtain the final score of a paper it is necessary to average out the scores given by all the participants. It is in this "global" averaging process that the relative expertise of the participants come in to play. Greater the expertise higher the weights, and like in the local case, this too is decided through a process of algorithmically defined consensus. This chapter deals with a principled scheme for obtaining such consensus driven scores and rankings.

The task has multiple real-world applications. For instance it is not uncommon for a participant of multiple online social networks (such as Linkedin, Facebook, G+ and also different instant messenger networks such as Gtalk, Ymessenger, etc) to voluntarily map their possibly different identities in the different networks into a common one, using services like Openid. These common id's can be used to conceptually tie the different networks into a loose federation with common participants. This chapter suggests a way of computing social standing of the participants in such a federation, where each edge is labeled by the identity of the social network that the edge exists in. [1]

[1]To elaborate, Alice maybe connected to Bob through Linkedin and Facebook. Perhaps the first indicates that they are colleagues and the second that they are also personal friends. Different people may use different networks to organize their contacts into different roles. An engineer may use Linkedin for professional contacts whereas a musician may use Myspace for the same. The task of defining a person's social importance in this combined network is isomorphic to the toy conference example given before, with the identity of the social network acting as a proxy for different criteria.

Some social networks, for example Google+ allow assigning different, private and possibly overlapping roles to one's contacts, organizing it into several circles of contacts. The labels assigned to such circles are entirely unrestricted and thus are not comparable across users. A user may have an implicit importance weight attached to each circle that he or she may not be willing to divulge. These user assigned roles can also serve as different criteria for ranking. This chapter provides a framework to rank such users even when the weights are not available.

Consider the hyperlink graph consisting of the current blog posts of several blogs. This graph is highly dynamic in nature and the cross references are almost always tagged by keywords of the authors' choice. Pagerank based ranking on such a graph could benefit from averaging out of the fluctuations. This chapter suggests how.

As a final application consider search engines that use link analysis to rank pages. They also can benefit from taking into account the role that they think a particular link plays in the graph, for example, navigational, commercial, endorsement of content, topical description etc. Anchor text may be used to detect these roles. It might not be very clear what the weights on these roles should be. This chapter addresses how one may assign such weights in a unsupervised but principled way.

The model is designed to address several kinds of uncertainty that may arise when ranking in multigraphs. Although link analysis is a richly researched subject (Kleinberg, 1999b), (Brin and Page, 1998), the topic of how to achieve consensus under uncertainty has not received as much attention. The is an initial step in that direction.

Although one would like the ranking procedure to be as automated as possible it is often essential to have a mechanism to modify the results, for example to counter new types of spam. One possible corrective intervention could be to define a desired partial order among the vertices. Our approach also provides for this capability. In fact the local recommendations obtained may be exclusively in the form of partial orders (rather than rank-scores) that need to be aggregated and reconciled.

Notation: Vectors are denoted by bold lower case letters. The i_{th} component of the vector $\boldsymbol{x}$ is indicated by x_i. When suitable, we also indicate the *entire* vector $\boldsymbol{x}$ by decorating its i^{th} component as follows: $\vec{x_i}$. This form is used to convey succinctly how a vector has been constructed from its components. Probability distributions used in this chapter are discrete and also denoted by bold lower case letters, with the letters $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{\rho}$ reserved for them. The symbol $T^\dagger$ indicates the transpose of matrix T. Random variables are also indicated by capital letters. $\mathop{\mathbb{E}}_{X\sim\boldsymbol{p}}[f(X)]$ represents the expectation of a function $f(\cdot)$ of a random variable X having a distribution $\boldsymbol{p}$. Sets are denoted by (matching) calligraphic letters, for instance random variable X takes values in a set $\mathcal{X}$. The unit simplex is denoted by Δ, its dimensionality will be implicit. For the most part we deal only with sets in the Euclidean vector space $\mathbb{R}^d$. The notation ${\mathbb{R}_+}^d$ will denote the positive orthant of $\mathbb{R}^d$, and $\mathbb{R}^d_\epsilon$ will denote the set $\{\boldsymbol{x}|\boldsymbol{x} \in \mathbb{R}^d \cap x_i > \epsilon\ \forall_i\}$, whereas the symbol Δ_ϵ will indicate the set $\{\boldsymbol{x}|\boldsymbol{x} \in \Delta \cap x_i > \epsilon\ \forall_i\}$ and the symbol $\blacktriangle$, the set $\{\boldsymbol{x}|\sum_i x_i \leq 1\ \boldsymbol{x} \in \mathbb{R}_+\}$

Basic knowledge of convex analysis is assumed. Interior, boundary and closure are denoted by int, bd and cl respectively, these are defined in terms of the native metric topology. The only exception is for non-empty domains of functions that have empty interiors in the native metric topology, in such cases we will consider the relative interior. The relative interior is the topological space defined by intersection of open sets in the native metric topology and the affine hull of the domain. $\mathrm{ConvHull}(\cdot)$ and $\mathrm{Extr}(\cdot)$ denote the convex hull and the extreme points respectively.

In order to reduce the proliferation of symbols some are re-purposed. For example, decoration with a $*$ when applied to functions indicate the Legendre conjugation operation, whereas when applied to variables denote some notion of optimality. With some abuse of notation we will indicate the set of limit points of the minimizing sequence $\boldsymbol{x}_t$ of the function $f(\boldsymbol{x})$ by $\mathrm{ArgInf}\, f(\boldsymbol{x})$, that is, for all sequences $\boldsymbol{x}_t$ with $\lim_{t\to\infty} \boldsymbol{x}_t = \mathrm{ArgInf}\, f(x)$ we have $\lim_{t\to\infty} f(\boldsymbol{x}) = \inf f(\boldsymbol{x})$. This just a notational convenience, there may not exist an *argument* which achieves the inf.

Since we deal with Markov chains as well as optimization, there is an unfortunate clash in terminology: "stationary point" is used to denote both a point where the cost function (or its Lagrangian) has a zero-gradient as well as a distribution that stays invariant under a Markovian transition. To alleviate the potential confusion we will use the term "0-gradient" point in the optimization setting.

5.0.1 Contributions

In the chapter we try to answer "what is the analogue of pagerank in the scenario where there is uncertainty over the edge weights of the (multi) graph?" That this is an important problem is motivated in the introduction with several applications. The original pagerank formulation is ill-equipped to provide an answer because it does not optimize any function. To mitigate this, the chapter

- Obtains pagerank as a solution of an optimization problem whose cost function penalizes deviation of "local ranks" from the "consensus" rank.
- It establishes that pagerank may be obtained by minimizing such deviance from consensus iff the cost function has the particular Bregman divergence form.
- The chapter provides algorithms that can be extended to the noisy multi-graph case and
- These iterative algorithms have simple and parallelizable updates that do not require any onerous synchronization or locking.

5.1 Preliminaries

In this preparatory section we review pagerank. For readers who are familiar with the background, this only serves to introduce notation. Subsequently, we give a mathematical formulation of our general problem, albeit at a high level, the specifics of which are solved

in the rest of the chapter. The problem include points of view that are both geometric as well as information theoretic. We focus on the geometric view.

Pagerank: Some algorithms, such as pagerank (Brin and Page, 1998) and HITS (Kleinberg, 1999b), rank vertices of a directed graph $\mathcal{G}$ by mapping the vertices in $\mathcal{V}$ to $\mathbb{R}$. They view $\mathcal{G}$ as a distributed recommendation system where each vertex recommends other vertices through its out-edges (directed edges that leave the vertex). In pagerank the *local recommendations* by a vertex v_i is represented as a $|\mathcal{V}|$ dimensional vector $\boldsymbol{t}_i$, whose j^{th} component denotes the strength of recommendation of vertex v_j by vertex v_i. The objective then is to obtain a global rank-score.

A global rank-score may be obtained from the local scores by combining them. A simple strategy is to use a convex combination, provided that the weights of combination are known. Uniform weighting, although a possibility, is unjustified because it is not consistent with the notion that vertices are inherently of unequal rank [2]. Thus, it is natural to seek weights of combination that are some monotonic increasing function of the global rank-score that it defines. The simplest relation between the weights and the global rank is the identity function. This yields pagerank, provided the local recommendation vectors are non-negative and L_1 normalized.

Pagerank can also be viewed as the stationary distribution of a Markov chain that traverses the underlying graph by following outlinks uniformly at random with occasional jumps to a random vertex. These two modes of traversal are chosen independently at each step, with probabilities α and $1-\alpha$. The second mode of traversal called "teleportation" serves as a mechanism to ensure that the chain is aperiodic and ergodic even when the underlying graph is not connected or acyclic.

Let A be the adjacency matrix of the graph and D_{out} be the diagonal matrix of its out-degrees, S a row stochastic "teleportation matrix", usually taken to be $\frac{1}{N}(\mathbf{1}\times\mathbf{1}^{\dagger})$, where $\mathbf{1}$ is a column vector of ones. The transition matrix of the pagerank equivalent Markov chain

[2] otherwise we would not be interested in ranking them.

is

$$T = \alpha \times D_{out}^{-1} \times A + (1 - \alpha) \times S.$$

From the property of aperiodic ergodic Markov chains it follows that the pagerank is uniquely determined for any $0 \leq \alpha < 1$ and that the pagerank iteration

$$r_i^{t+1} = \alpha \sum_{j \in \mathcal{N}_i} r_j \boldsymbol{t}_{\boldsymbol{j}_i} + (1 - \alpha) \frac{1}{N} \tag{5.1}$$

converges to the primary eigenvector $\boldsymbol{\rho}$ of $T^{\dagger}$, the stationary probabilities of the Markov chain.

Outline of Divergence Based Consensus Ranking Problem:

Keeping in view the pagerank approach, let us introduce the proposed divergence based formulation used to solve the general consensus ranking problem. We skip over a lot of detail as this is intended to familiarize the reader with the high-level features of the underlying mathematical model. The finer details are filled in due course.

Consider a subset $\mathbb{S} \subset \mathbb{R}^n$ and a distance like divergence function $D(\cdot, \cdot) : (\mathbb{S}, \mathbb{S}) \mapsto \mathbb{R}_+$ that only satisfies the requirement $D(\boldsymbol{x}, \boldsymbol{y}) = 0 \iff \boldsymbol{x} = \boldsymbol{y}$. Following the pagerank interpretation that vectors $\boldsymbol{t}_i$ represent the "local" recommendations by the i^{th} vertex, a constructive definition of a consensus rank-score vector is a vector $\boldsymbol{r}$ that is closest to all such local recommendations $\boldsymbol{t}_i$. If however, the recommenders were to provide only the sets of uncertainty $\boldsymbol{T}_i$ in which their rank score vector $\boldsymbol{t}_i$ lies, the consensus may be defined as:

$$\begin{aligned} \boldsymbol{r}^*(\boldsymbol{w}) &= \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \sum_i w_i D(\boldsymbol{t}_i, \boldsymbol{r}) \\ &= \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle \,. \end{aligned} \tag{5.2}$$

The weight vector $\boldsymbol{w}$ is a parameter that needs to be chosen. Here, we take inspiration from

pagerank's justification of the fixed point definition and choose $\boldsymbol{w}$ that satisfies

$$\boxed{\boldsymbol{r}^*(\boldsymbol{w}) = \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle = \boldsymbol{w},} \tag{5.3}$$

Brouwers fixed point theorem guarantees that there is at least one such fixed point, however, there is likely to be many, all of which are equivalent in terms of (5.3). In the event of multiple solutions one can take an optimistic view or a pessimistic view, where one chooses the fixed point that achieves the minimum distance

$$\boxed{\begin{aligned} &\min \langle \boldsymbol{w}, D(\boldsymbol{t}_i^*, \boldsymbol{w}) \rangle \\ &\quad \text{s.t.} \quad \boldsymbol{w} = \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle \\ &\quad \boldsymbol{t}_i^* = \min_{\boldsymbol{r}} \text{Argmin}_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle \end{aligned}} \tag{5.4}$$

or one that chooses the fixed point that achieves the maximum distance

$$\boxed{\begin{aligned} &\max \langle \boldsymbol{w}, D(\boldsymbol{t}_i^*, \boldsymbol{w}) \rangle \\ &\quad \text{s.t.} \quad \boldsymbol{w} = \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle \\ &\quad \boldsymbol{t}_i^* = \min_{\boldsymbol{r}} \text{Argmin}_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle . \end{aligned}} \tag{5.5}$$

Specializations of (5.4) and (5.5) are the central problem that we solve in this chapter. It should be readily apparent that for arbitrary divergence function D, equation (5.3) is a difficult, non-linear, implicitly defined and a cumbersome fixed point problem. *One key difficulty is that the function $\boldsymbol{r}^*(\boldsymbol{w})$, defined in* (5.2)*, whose fixed point we seek, is not known in closed form, but specified (variationally) as an optimization problem*. The obvious questions that crop up are:

- whether there exists a solution
- whether the solution is unique

- are there algorithms that provably converge to these fixed points from arbitrary initialization
- how fast do these algorithms converge.

We cannot address these questions for all divergence functions D. We study specializations that can be solved tractably with provable convergence guarantees. We claim that if we restrict $\boldsymbol{r}$ to the set $\mathbb{S} \subset \{\boldsymbol{x} | \sum_i x_i = c\}$, where c is an arbitrary constant, the family of Bregman divergences are the only choice for D such that for every choice of $\boldsymbol{t}_i \in \mathbb{S}$ the $\boldsymbol{r}$-minimization sub-problem (5.3) reduces to a linear eigen-problem. The guaranteed existence of eigenvalues will play an important role in the algorithm proposed.

For the special case of (Legendre) Bregman divergences defined by "essentially smooth" convex functions, it is quite surprising that we can solve (5.5) by dropping the fixed point constraint. The constraint is automatically satisfied at the optimum. We cannot emphasize it enough that this simplifies the cumbersome, variationally specified fixed point problem into a much simpler optimization problem.

Before considering the problem in full generality of Bregman divergences, we introduce the details by considering a specific member: KL divergence. The algorithms work almost word for word for any Bregman divergence defined on $\mathbb{S}$ without incurring much additional complexity, allowing a practitioner to tailor the choice to an application.

We generalize to Bregman divergences in section 5.3 and finally generalize to the consensus ranking problem over sets T_i in section 5.4.1. In section 5.3 we present some new results concerning Bregman divergences that are vital to the derivation of the updates that are used in the ranking algorithm. The scope of these new results are wide enough to be of independent interest.

5.2 Pagerank as Consensus Over *Vectors*

As indicated in Section 5.1 we will pursue two different optimization theoretic routes to consensus ranks, one that will correspond to equation (5.4) and the other to equation (5.5). Both these formulations are designed to handle sets of uncertainty T_i, providing a way to obtain consensus rank score vector $\boldsymbol{\rho}$ from the sets T_i. Rather than discussing consensus over sets right away, we build up gradually by considering consensus over vectors $\boldsymbol{t}_i$. In other words, initially we treat $T_i = \{\boldsymbol{t}_i\}$ to be singletons to show that pagerank is naturally recovered. This will clarify that the two routes are alternative generalizations of pagerank. A key idea is to demonstrate that we are able to shed the fixed point baggage entirely, and pose pagerank as an optimization problem. This will simplify the approaches (5.4) and (5.5) significantly.

Quite remarkably, if we optimize the functions with the fixed point set constraint removed, under conditions, the fixed point condition is automatically recovered at the optimum. This lets us convert a difficult variationally specified fixed point problem into an optimization or a saddle point problem.

To solve (5.4) specialized to KL divergence and singleton T_is we provide a *conceptual* algorithm that converges to the global minimum. Further we show that pagerank is the limit point of this conceptual algorithm. This establishes that pagerank is indeed the global minimum of unconstrained (5.4). However the cost function is not convex and may have more than one minimum and the conceptual algorithm requires the global minimum be obtained. In contrast, we provide a simple *realizable* alternating minimization algorithm parameterized by a penalty parameter β that in the limit converges to the local minimum of the cost function, and for finite β obtains the local minimum of an arbitrarily tight lower bound. As an alternative to the Min-Min, alternating minimization formulation we reduce (5.5) to a Max-Min saddle point problem by replacing the complicated fixed point constraint by a simple nested unconstrained minimization over another auxiliary variable.

Thus pagerank is posed as the outcome of two separate optimization problems that

differ in the degrees of convenience and generality offered. One of them uses a Min-Min formulation, the other a Min-Max. The merits and demerits are summarized in table 5.1. The curious reader may skip ahead and consult it, however, for full appreciation, familiarity with the algorithms developed in sections 5.3.1 and 5.3.2 is necessary.[3]

5.2.1 Kullback Informatic, Optimistic Consensus Over *Vectors*

As the first contribution we provide a novel cost function based view of pagerank. The cost is directly motivated by a notion of rank/consensus quality and will serve as a stepping stone in our path to a solution of the consensus ranking problem. Recall that pagerank (Brin and Page, 1998) was originally defined directly as the fixed point of an update, there were no cost functions involved.

Although one may directly change the functional form of the pagerank updates, that would be ad hoc. One also has to be careful so as not to disrupt the guarantees of convergence. Rather than follow this route, we identify functions that pagerank is a minimizer or a saddle point of. Once obtained, we add extra terms to that function to capture the requirements of consensus. [4]

Pagerank, An Alternative View:

Recall that the recommendation graph $\mathcal{G}$ has outlinks that can be interpreted as a local recommendation of the edge recipient by the donor vertex. The local recommendation of vertex v_i is represented by an ℓ_1 normalized vector $\boldsymbol{t}_i$ of dimension $|\mathcal{V}|$. The weight of

[3] Both the *Min-Min* and *Min-Max* optimization formulations presented lead to corresponding solutions of the unsupervised consensus ranking problem over sets T_i. They differ in how the fixed point property is achieved (by penalization in the first and by saddle point in the second) and what guarantees they provide.

[4] Formulations that only penalize the deviation from pagerank-stationarity, e.g. $\min_{\boldsymbol{\rho}} \min_{\boldsymbol{t}_i \in \boldsymbol{T}_i} \mathrm{KL}\left(\sum_i \rho_i \boldsymbol{t}_i \| \boldsymbol{\rho}\right)$ or $\max_{\boldsymbol{\rho}} \max_{\boldsymbol{t}_i \in \boldsymbol{T}_i} \left\langle \boldsymbol{\rho}, [\boldsymbol{t}_1 \cdots \boldsymbol{t}_i \cdots \boldsymbol{t}_{|\mathcal{V}|}] \boldsymbol{\rho} \right\rangle$ are unsatisfying because they cannot distinguish between multiple vectors that achieve pagerank-stationarity and does not offer interpretation as a ranking quality measure.

this donor vertex is w_i. By pagerank convention

$$\boldsymbol{t}_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_i|} & \text{if } v_i \text{ recommends } v_j \\ 0 & \text{otherwise.} \end{cases}$$

The symbol $\mathcal{N}_i$ represents the set of out-neighbors of the vertex v_i. The normalized vectors are stacked to form a matrix T such that the i^{th} row $T(i,\cdot)$ is $\boldsymbol{t}_i{}^\dagger$, just as discussed in section 5.1.

The optimal consensus can be defined as the vector $\boldsymbol{\rho}$ closest in KL sense to the recommendations of all the vertices weighted by their importance $\boldsymbol{w} \in \Delta$. A regularizing term enforces that $\boldsymbol{\rho}$ is close to $\boldsymbol{s}$, a prior rank vector, usually taken to be uniform, and α is a parameter in $(0,1)$. This leads to the cost function

$$F(\boldsymbol{w},\boldsymbol{\rho}) = \alpha \sum w_i \mathrm{KL}\left(\boldsymbol{t}_i \| \boldsymbol{\rho}\right) + (1-\alpha)\mathrm{KL}\left(\boldsymbol{s}\|\boldsymbol{\rho}\right). \tag{5.6}$$

The vector $\boldsymbol{s}$ and the parameter α play the same role as the jump probabilities in the original pagerank formulation. For any choice of $\boldsymbol{w} \in \Delta$, the global minimum is given by the weighted average

$$\boldsymbol{\rho}_*(\boldsymbol{w}) \triangleq \mathrm{Argmin}_{\boldsymbol{\rho}}\, F((\boldsymbol{w},\boldsymbol{\rho})) = \alpha \sum_i w_i \boldsymbol{t}_i + (1-\alpha)\boldsymbol{s}. \tag{5.7}$$

Comparing equations (5.1) and (5.7) one can observe that pagerank formulation follows if the weights $\boldsymbol{w}$ happen to be identical to the consensus ranks, i.e. if $\boldsymbol{w} = \boldsymbol{\rho}_*(\boldsymbol{w})$. The reader will note that this is exactly the condition (5.3). We will refer to this condition as *pagerank stationary condition*.

Mean-ArgMin Coincidence: The coincidence of the minimum and the mean in (5.7) is a consequence of a more general result involving Bregman divergences, of which KL divergence is a special case. The general result is presented as theorem 17 (see Ap-

pendix) and can be used to prove the useful expansion

$$F(\boldsymbol{w}, \boldsymbol{\rho}) = F(\boldsymbol{w}, \boldsymbol{\rho}_*(\boldsymbol{w})) + \mathrm{KL}\left(\boldsymbol{\rho}_*(\boldsymbol{w}) \| \boldsymbol{\rho}\right). \tag{5.8}$$

One may try to ensure *pagerank stationarity* by adding it as a constraint, yielding:

$$\mathrm{Min}_{\boldsymbol{\rho},\boldsymbol{w}}\, F(\boldsymbol{w}, \boldsymbol{\rho}) \quad \text{s.t.} \quad \boldsymbol{w} = \boldsymbol{\rho}, \text{ and } \boldsymbol{w} \in \Delta. \tag{5.9}$$

Note, the expression for $\boldsymbol{\rho}_*(\boldsymbol{w})$ in equation (5.7) is for the unconstrained case and hence the form need not apply for the constrained case (5.9). One may search for the minima of (5.9) directly by eliminating the constraint in (5.9) by substitution, to yield:

$$\underset{\boldsymbol{\rho}}{\mathrm{Min}} G(\boldsymbol{\rho}) = \underset{\boldsymbol{\rho}}{\mathrm{Min}}\, \alpha \sum \rho_i \mathrm{KL}\left(\boldsymbol{t}_i \| \boldsymbol{\rho}\right) + (1-\alpha)\mathrm{KL}\left(s \| \boldsymbol{\rho}\right). \tag{5.10}$$

Let $\boldsymbol{\rho}^*$ be the solution of the problem (5.9) or equivalently (5.10). An important question is whether $\boldsymbol{\rho}^*$ satisfies *pagerank stationarity*. The behavior of the constraint in (5.9) at $\boldsymbol{\rho}^*$ is critical to this stationarity question.[5] The cost function is not convex and may have multiple local minimum, however, as we shall show, the global minimum satisfies *pagerank stationarity*. The demonstration will be a little elaborate because the conventional tools are not well suited to analyze properties of global optimum. We shall introduce a penalty based algorithm in section 5.2.2 that on one hand solves (5.10) and provides a proof of pagerank stationarity on the other.

[5]Consider evaluating the function G at $\boldsymbol{\rho}^*$ which is equivalent to evaluating $F(\cdot,\cdot)$ at $(\boldsymbol{w} = \boldsymbol{\rho}^*, \boldsymbol{\rho} = \boldsymbol{\rho}^*)$. Let us restrict $\boldsymbol{w} = \boldsymbol{\rho}^*$ and relax the constraint on $\boldsymbol{\rho}$ present in 5.9. Now if we re-optimize over the free variable $\boldsymbol{\rho}$ and had the constraint been active at $(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$, the minima would shift to $(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))$ and violate pagerank stationarity. In this hypothetical case pagerank stationarity will not hold. On the other hand if the constraint is inactive then pagerank and $\boldsymbol{\rho}^*$ will coincide.

5.2.2 Min-Min Coordinate Descent Formulation

A deliberate and a persistent motif in this chapter is optimization through closed form coordinate-wise updates. The coupling of the variables ρ_i, ρ_j in (5.10) makes it difficult, therefore we work with function $F(\cdot,\cdot)$ (5.9), where the variables are uncoupled (except in the constraint). Our interest lies only in the feasible set $(\boldsymbol{w} = \boldsymbol{\rho})$ of the domain of $F(\cdot,\cdot)$. In order to focus on that region, we add a sequence of increasing penalty terms that is active everywhere outside of the constraint set, and optimize this sequence of unconstrained, and hence, decoupled cost functions by alternating minimization updates. The relation between the cost functions is shown in figure 5.1 and figure 5.2.

We must, however, choose the form of the penalty function carefully to maintain closed formed nature of the updates and it will also be critical in showing that pagerank stationarity is retained.

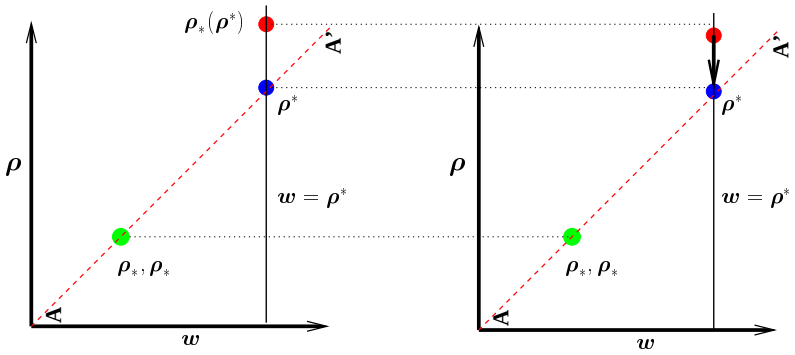


Figure 5.1: Left: The red line AA' denotes the constraint $\boldsymbol{w} = \boldsymbol{\rho}$. The pagerank is $(\boldsymbol{\rho}_*, \boldsymbol{\rho}_*)$ and an arbitrary solution to problem (5.9) is $(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$. If the constraint in problem (5.9) is relaxed the optima shifts from $(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$ to $(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))$. Right: We add a penalty term that is active everywhere outside the constraint set AA' by adding sufficient penalty we may increase the value at $(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))$ to be greater than $(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$ and hence move the minima towards it. Significantly enough, for the Penalty based optimization it converges to $(\boldsymbol{\rho}_*, \boldsymbol{\rho}_*)$.

There are two important choices to be made: (i) the functional form of the penalty term and (ii) the weight assigned to it. In the next few paragraphs we explain our choices.

One advantage of using KL divergence in expression (5.9) is the fact that the uncon-

strained optimizer is available in closed form and that the form of the minimizer is linear in $\boldsymbol{t}$. Both these properties are important, the former is a convenience whereas the latter is *essential* in the reduction of pagerank-stationarity to a linear eigen-problem. Hence it is important that the penalty term we use also preserves the closed form and the linearity of the minimizer. We show that both can be achieved by using a (i) penalty term that is based on the same divergence that is used in the unpenalized form, i.e. the KL divergence and (ii) for a particular choice of the left right order of the arguments. Later we show that this holds true for a larger class known as Bregman divergence and more critically that Bregman divergences are the *the only* class for which the reduction to a linear eigen-problem is possible.

For generality and convenience, we absorb the parameters α and $\boldsymbol{s}$ into modified distributions $\hat{\boldsymbol{t}}_i$, and define a associated cost function that is a valid surrogate for (5.9), as follows

$$\hat{\boldsymbol{t}}_i \triangleq \alpha \boldsymbol{t}_i + (1-\alpha)\boldsymbol{s} \text{ and } \hat{F}(\boldsymbol{w}, \boldsymbol{\rho}) \triangleq \sum w_i \mathrm{KL}\left(\hat{\boldsymbol{t}}_i \| \boldsymbol{\rho}\right). \tag{5.11}$$

Note that the optimality of $\boldsymbol{\rho}_*(\boldsymbol{w})$ and the correspondence with pagerank update are preserved for $\boldsymbol{w} \in \Delta$. This transformation has another consequence, now each component of $\hat{\boldsymbol{t}}_i$ can be bounded below by $(1-\alpha)\min_i s_i$. [6]

Penalty Method Formulation

The optimization problem with the penalized cost function is the following:

$$\hat{F}(\boldsymbol{w}, \boldsymbol{\rho}) + \frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{w} \| \boldsymbol{\rho}\right), \quad 0 \leq \beta \leq 1,\ \boldsymbol{w} \in \Delta. \tag{5.12}$$

It takes the same value as $\sum_i \rho_i \mathrm{KL}\left(\hat{\boldsymbol{t}}_i \| \boldsymbol{\rho}\right)$ on the set $\boldsymbol{\rho} = \boldsymbol{w}$. Outside of this set the cost function is penalized by $\frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{w} \| \boldsymbol{\rho}\right)$, smaller the value of β higher is the penalization. Expression (5.12) can be minimized over $\boldsymbol{\rho}$ and $\boldsymbol{w}$ using the updates

[6] This boundedness will turn useful later in ensuring progress towards constraint satisfaction, in particular as a consequence of lemma 18 and 19, to be introduced shortly.

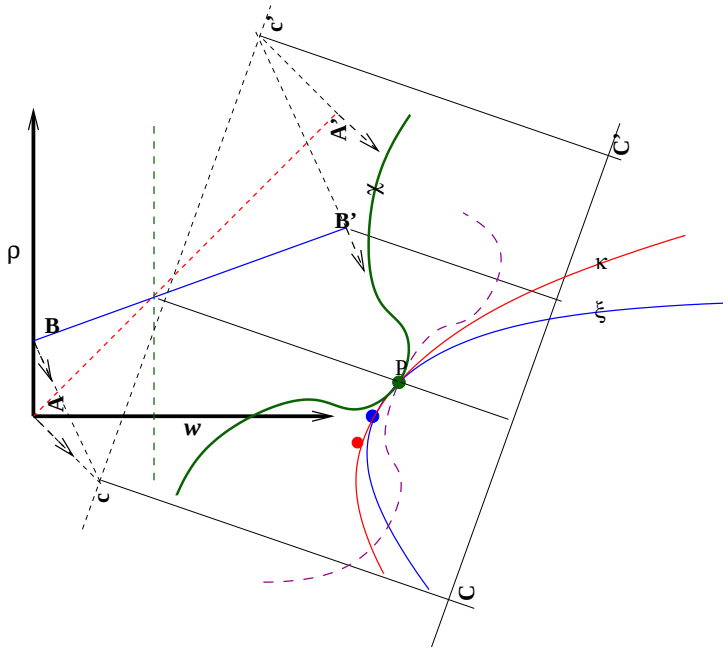


Figure 5.2: Plots of the cost function on different sections of the product space $\boldsymbol{w} \times \boldsymbol{\rho}$. AA'(in red) defines the constraint set $\boldsymbol{w} = \boldsymbol{\rho}$, BB'(in blue) defines the set $\boldsymbol{\rho} = \boldsymbol{\rho}_*(\boldsymbol{w})$. The minimum cost function along these sections are scaled to a common X-axis cc', alternatively CC'. Plot χ (in green) indicates $\sum_i w_i \mathrm{KL}\left(\boldsymbol{t}_i \| \boldsymbol{\rho}\right)$ for different $\boldsymbol{\rho}$ with $\boldsymbol{w}$ fixed at the stationary value (in dotted green). It achieves a unconstrained global optima at P. The function values are tracked for different values of $\boldsymbol{w}$ for the two sections (i) the constrained set AA' to give curve κ (in red) and (ii) BB' the set of unconstrained optima $\boldsymbol{\rho}_*(\boldsymbol{w})$ to give the curve ξ (in blue), upper bounded by κ and tight at point P. Curves ξ and κ envelop the optimal point of the cost function over the constrained set BB' and AA'. The optima of the curves χ, ξ and κ are indicated by points colored, green, blue and red.

$$
\begin{aligned}
{\boldsymbol{\rho}_*}^{t+1}(\beta, \boldsymbol{w}^t) &= \operatorname*{Argmin}_{\boldsymbol{\rho}} \hat{F}(\boldsymbol{w}^t, \boldsymbol{\rho}) + \frac{1-\beta}{\beta} \mathrm{KL}\left(\boldsymbol{w}^t \| \boldsymbol{\rho}\right) \\
&= \beta \left(\alpha \sum_i \boldsymbol{w}_i \boldsymbol{t}_i + (1-\alpha)\boldsymbol{s} \right) + (1-\beta)\boldsymbol{w}^t \text{ and} \qquad (5.13) \\
w_i^{t+1}(\beta, {\boldsymbol{\rho}_*}^{t+1}) &= \operatorname*{Argmin}_{w_i, \boldsymbol{w} \in \Delta} \hat{F}(\boldsymbol{w}, {\boldsymbol{\rho}_*}^{t+1}) + \frac{1-\beta}{\beta} \mathrm{KL}\left(\boldsymbol{w} \| {\boldsymbol{\rho}_*}^{t+1}\right) \\
&\propto \rho_i^{t+1} e^{-\frac{\alpha\beta}{1-\beta}\left(\mathrm{KL}\left(\hat{\boldsymbol{t}}_i \| {\boldsymbol{\rho}_*}^{t+1}\right) - \lambda\right)}. \qquad (5.14)
\end{aligned}
$$

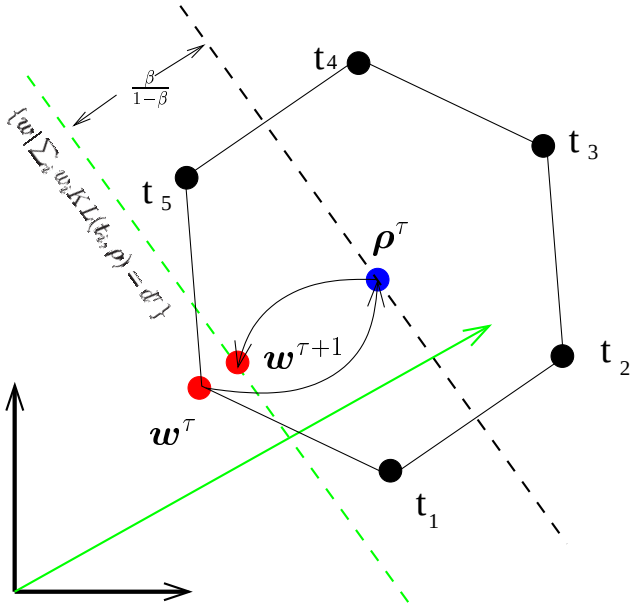


Figure 5.3: The penalty based updates: The estimate of the rank vector $\boldsymbol{\rho}^T$ (shown in blue) in the T^{th} iteration is computed in the $\boldsymbol{\rho}$ *update* (5.13) as a weighted mean of the vectors $\boldsymbol{w}^T$ (shown in red) and $\hat{\boldsymbol{t}}_i$ (equation (5.11)) with weights $(1-\beta)$ and βw_i respectively. In the subsequent step, given by equation (5.14), $\boldsymbol{w}$ is updated by KL(or more generally Bregman) projecting $\boldsymbol{\rho}^T$ on the updated hyperplane (shown in green) defined by $\{\boldsymbol{w}|\left\langle \vec{w}_i, \mathrm{KL}\left(\hat{\boldsymbol{t}}_i \| \boldsymbol{\rho}^T\right)\right\rangle = d^T\}$, such that the symmetrized Bregman divergence between $\boldsymbol{\rho}^T$ and $\boldsymbol{w}^{T+1}$ is $\frac{\beta}{1-\beta}$ times their Euclidean distance along the normal to the hyperplane.

The superscript t indicates the iteration counter and λ in (5.14) imposes the normalization condition. Both the updates are depicted geometrically in figure 5.3. Update (5.13) is a weighted mean of $\boldsymbol{t}_i, \boldsymbol{s}, \boldsymbol{w}$. Update (5.14) is explored in more detail in Section 5.3.1, it is an I-projection (see section 5.1) of the vector $\boldsymbol{\rho}$ on the hyperplane defined by the normal direction $\overrightarrow{\mathrm{KL}\left(\hat{\boldsymbol{t}}_i \| \boldsymbol{\rho}\right)}$ and such that the projection $\boldsymbol{w}$ on the hyperplane is at an I-divergence of $\frac{\beta}{1-\beta}$ times its Euclidean distance from the vector $\boldsymbol{\rho}$.

Note that the cost function (5.12) is continuously differentiable, strictly convex in $\boldsymbol{\rho}$ and $\boldsymbol{w}$ separately, and the alternating minimizers (5.13, 5.14) are uniquely achieved. Thus it follows (Bertsekas, 1999) that iterations of alternate minimizers converge to the 0-gradient point of the cost function (5.12) (although, not necessarily to the global optimum

of (5.12)). It also follows that the global optimum of (5.12) satisfies the coordinate-wise optimum relations (5.13), (5.14) for any finite β.

Using standard results for penalty based methods one can show that if the globally optimal $\boldsymbol{w}, \boldsymbol{\rho}_*$ is obtained for every β and $\beta \to 0$ (i) the constraint $\boldsymbol{w} = \boldsymbol{\rho}_*$ is achieved in the limit and (ii) $\boldsymbol{w}, \boldsymbol{\rho}_*$ achieves the global optimum of (5.10). Since the constraint is achieved in the limit, from (5.13) we obtain optimal $\boldsymbol{w} = \alpha \sum_i \boldsymbol{w}_i \boldsymbol{t}_i + (1-\alpha)\boldsymbol{s}$ in the limit, which is in exact equivalence with pagerank. We will make our arguments more formal when we rephrase the method in the full generality of Bregman divergences in Section 5.3.1. The updates (5.13, 5.14) help in proving that pagerank is the optimizer of (5.12) but does not necessarily guarantee that this will be reached, because optimality is not guaranteed by the updates (5.13, 5.14), only convergence to a 0-gradient point is.

Lacking convexity in (5.9), satisfying the necessary KKT conditions is the best that one can realistically aim for. Do the updates converge to a point satisfying the necessary KKT conditions of (5.9) ? For penalty methods where convergence is guaranteed only to a potentially non-optimal 0-gradient point, the convergence to a KKT satisfying point is in general not guaranteed in the limit. So what can one claim off ${\boldsymbol{\rho}_*}^{\infty}(\beta, \boldsymbol{w}^{\infty}), \boldsymbol{w}^{\infty}(\beta, {\boldsymbol{\rho}_*}^{\infty})$ as $\beta \to 0$ in this case ? We shall show that for strongly convex Bregman divergences defined on a bounded domain, one can guarantee convergence to a 0-gradient point of an arbitrarily tight lower bound of (5.10).

Now we explore, how the magnitude of β affects the accuracy of our solution. We present a couple of lemmas that sheds some light on the question.

Lemma 18. *For two discrete distributions $\boldsymbol{p}$ and $\boldsymbol{q}$ such that $\min_i p_i \geq \epsilon$ and $\min_i q_i \geq \epsilon$ the ratio of the forward and the backward KL divergence $\frac{\mathrm{KL}(\boldsymbol{p}\|\boldsymbol{q})}{\mathrm{KL}(\boldsymbol{q}\|\boldsymbol{p})}$ is bounded above by $\frac{2}{\epsilon}$.*

proof: See appendix.

Lemma 19. *Let $\min_{ij} \hat{\boldsymbol{t}}_{ij} > \epsilon$ and let $\boldsymbol{\rho}^*$ the minimum of* (5.10) *also satisfy $\min_i \rho_i^* > \epsilon$. Consider any point $(\boldsymbol{\rho}^*, \tilde{\boldsymbol{\rho}})$ lying between $(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$ and $(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))$ and satisfying $\frac{\|\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) - \tilde{\boldsymbol{\rho}}\|}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|} \leq$*

$\frac{\delta}{1-\delta}$ *for a fixed* $\delta \in (0,1)$. *For the penalized cost function* (5.12) *evaluated at* $(\boldsymbol{\rho}^*, \tilde{\boldsymbol{\rho}})$ *to be higher than* $F(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*)$ *it is sufficient that* $\frac{1-\beta}{\beta} \geq \frac{1}{\epsilon}(2 + \frac{\delta}{(1-\delta)})$.

proof: See appendix.

Lemma 19 allows us to control the proximity of the optimum of the penalized optimization problem (5.12) to the desired set of $\boldsymbol{w} = \boldsymbol{\rho}$. Ideally we would require that $\delta = 1$ which would then satisfy the *pagerank stationarity condition* exactly, however, in this case the required β becomes unbounded. We can however choose β so that the solution of (5.12) is arbitrarily close.

For lemma 19 to be applicable we require $\rho_i \geq \epsilon$. If we choose $(1-\alpha)\min_i s_i \geq \epsilon$ the $\boldsymbol{\rho}$ updates in equation (5.13) maintains the bound $\rho_i \geq \epsilon$ provided $\boldsymbol{w} \in \Delta_\epsilon$, though the $\boldsymbol{w}$ update (5.14) need not. However with a minor modification to update (5.14) we can ensure $\boldsymbol{w} \in \Delta_\epsilon$. Let $\mathbb{R}^d_\epsilon = \{\boldsymbol{x} | \boldsymbol{x} \in \mathbb{R}^d \cap x_i > \epsilon \; \forall_i\}$. Consider the modification

$$\hat{F}(\boldsymbol{w}, \boldsymbol{\rho}) + \frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{w} \| \boldsymbol{\rho}\right), \quad 0 \leq \beta \leq 1, \quad \boldsymbol{s}, \boldsymbol{w} \in \Delta \cap \mathbb{R}^d_\epsilon. \tag{5.15}$$

The updates corresponding to (5.13) remain unchanged but those corresponding to (5.14) changes to

$$w_i \propto \rho_i e^{-\lambda_i \frac{\alpha\beta}{1-\beta}(\mathrm{KL}(\boldsymbol{t}_i \| \boldsymbol{\rho}) - \lambda)}, \tag{5.16}$$

where the Lagrange multipliers λ_i have to be determined (numerically) such that the constraint $\boldsymbol{w} \in \Delta \cap \mathbb{R}^d_\epsilon$ is satisfied. Now notice that all preconditions of lemma 19 are satisfied, hence we can state the following theorem:

Proposition 4. *It is sufficient to set* $\frac{1-\beta}{\beta} \geq \frac{1}{\epsilon}(2 + \frac{\delta}{1-\delta})$ *in order to ensure that the minimum of* $\hat{F}(\boldsymbol{w}, \boldsymbol{\rho}) + \frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{w} \| \boldsymbol{\rho}\right), \quad 0 \leq \beta \leq 1, \quad \boldsymbol{s}, \boldsymbol{w} \in \Delta \cap \mathbb{R}^d_\epsilon$ *is obtained at* $\tilde{\boldsymbol{\rho}}$ *that satisfies the desired pagerank stationarity condition of* $\boldsymbol{w} = \boldsymbol{\rho}$ *with an arbitrary but bounded degree of proximity that is controlled by the relation* $\frac{\|\boldsymbol{\rho}_*(\boldsymbol{w}) - \tilde{\boldsymbol{\rho}}\|}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|} \geq \frac{\delta}{1-\delta} \; \forall \boldsymbol{w}$ *where* $\boldsymbol{\rho}_*(\boldsymbol{w})$ *is the unconstrained solution of the optimization problem* (5.6).

Proof. Follows directly from lemma 19. □

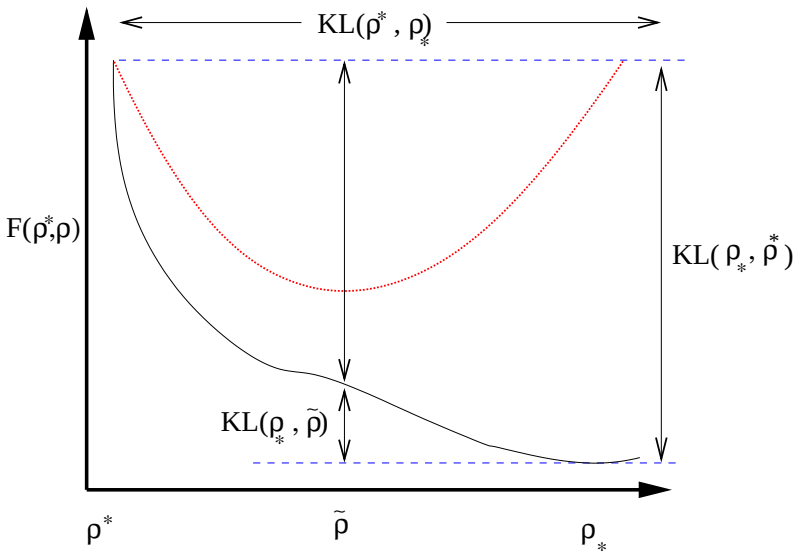


Figure 5.4: Shows a schematic view of the cost function (5.6) (in black) and the penalized cost function (5.12) (in red) for a fixed $\boldsymbol{w}$ set to $\boldsymbol{\rho}^*$. Though the figure refers to KL divergence, the schematic applies equally to the general Bregman divergence case as well. To represent this generality, the curves have been drawn to be non-convex. Bregman divergences may be non-convex in the second argument but KL divergence in particular is not. The point $\boldsymbol{\rho}_*$ represents the unconstrained minimum of (5.6) for a fixed value of $\boldsymbol{w}$, here set to $\boldsymbol{\rho}^*$. The fractions δ and $1-\delta$ are explained in the text.

5.3 Bregman Informatic Consensus over Vectors

In the rest of the chapter we generalize the ranking problem along two lines. One generalization is to consider Bregman divergences rather than KL divergences. This adds little or no complexity to the algorithm developed. On the other hand this allows a practitioner to choose a Bregman divergence appropriate for the application. All Bregman divergences used in this chapter will be defined over a bounded subset of the affine manifold $\{\boldsymbol{w} | \langle \mathbf{1}, \boldsymbol{w} \rangle = 1\}$, will be $1-$strongly convex and $\operatorname{dom} \phi^*$ will be all of $\mathbb{R}^n$. It can be easily verified that KL divergences satisfy these conditions.

The second generalization addresses our final goal, that of consensus ranking. This is achieved by considering local recommendations that are no longer restricted to be single

vectors but to convex sets of vectors in $\mathbb{R}^{|\mathcal{V}|}$ that expresses the uncertainty over the rankings.

Before developing these extensions, recalling relevant background in convex analysis and Bregman divergences is absolutely essential. We also present new results concerning properties of Bregman divergences that are not only interesting in their own right but critical to the formulation.

In the interest of keeping the flow cohesive, the background as well as new material concerning Bregman divergences have been moved to Appendix 2.2. We do want to remind the reader that it is essential to what follows and some of the contributions of this chapter lay there.

We start with a simple **recipe** to create Bregman divergences meeting the criteria mentioned above. Let $\{\boldsymbol{y}_i\}_{i=1}^n$ be the extreme points of a polytope defined on $\{\boldsymbol{w} | \langle \mathbf{1}, \boldsymbol{w} \rangle = 1\}$. Let $\phi(\boldsymbol{x}) = \sup_{\boldsymbol{\theta}} \langle \boldsymbol{x}, \boldsymbol{\theta} \rangle - \log(\sum_i^n \exp(\theta_i, y_i))$. The function $\phi(\boldsymbol{x})$ is strongly convex and thus can be scaled to yield a $1-$strongly convex function. Furthermore its domain is the convex hull of $\{\boldsymbol{y}_i\}_{i=1}^n$. Note that KL divergence can also be obtained by using this recipe and choosing $\{\boldsymbol{y}_i\}_{i=1}^n$ to be the vertices of the unit simplex.

5.3.1 Bregman Informatic, Optimistic Consensus over *Vectors*

The Bregman divergence based formulation is obtained by a direct replacement of the KL divergence in formulation (5.12) with a Bregman divergence. Redefining $\tilde{F}(\boldsymbol{w}, \boldsymbol{\rho})$ as $\tilde{F}(\boldsymbol{w}, \boldsymbol{\rho}) \triangleq \sum_i w_i D_\phi\big(\hat{\boldsymbol{t}}_i \big|\big| \boldsymbol{\rho}\big)$ we obtain the Bregman divergence based coordinate-wise Min-Min formulation:

$$\min_{\boldsymbol{w}, \boldsymbol{\rho}} \tilde{F}(\boldsymbol{w}, \boldsymbol{\rho}) + \frac{1-\beta}{\beta} D_\phi\big(\boldsymbol{w} \big|\big| \boldsymbol{\rho}\big) \text{ s.t. } \langle \boldsymbol{w}, \mathbf{1} \rangle = 1. \tag{5.17}$$

Consider the following conceptual[7] algorithm:

[7] It is conceptual because it requires minimization of a non-convex function

Initialize: Fix a series $c_t \to \infty$. Set $t = 0,\ \frac{1-\beta}{\beta} = c_t$.

Repeat: Till $\boldsymbol{w}_t = \boldsymbol{\rho}^t$

Compute: $\boldsymbol{w}^t, \boldsymbol{\rho}^t = \mathrm{Argmin}_{\boldsymbol{w},\boldsymbol{\rho}}\, \tilde{F}(\boldsymbol{w}, \boldsymbol{\rho}) + \frac{1-\beta}{\beta} D_\phi\big(\boldsymbol{w} \big|\big| \boldsymbol{\rho}\big)$ s.t. $\langle \boldsymbol{w}, \mathbf{1}\rangle = 1$

Set: $\frac{1-\beta}{\beta} = c_{t+1}$

Return: $\boldsymbol{\rho}$

Proposition 5. *Running the conceptual algorithm above one obtains*

- $\lim_{t\to\infty} \boldsymbol{w}^t \to \boldsymbol{\rho}^t$ *and*
- $\lim_{t\to\infty} \tilde{F}(\boldsymbol{w}^t, \boldsymbol{\rho}^t) \to \inf_{\boldsymbol{\rho}} \tilde{F}(\boldsymbol{\rho}, \boldsymbol{\rho}) \quad$ *s.t.* $\langle \boldsymbol{w}, \mathbf{1}\rangle = 1.$

Proof. Follows from specializing Theorem in Zangwill (1969). □

The joint Argmin step in the algorithm above is intractable because of lack of joint convexity. Thus, in the realizable algorithm we replace it by steps that achieve KKT necessity, by alternating minimization updates. As a consequence of theorem (17) part (C.3) described in Appendix 2.2 the $\boldsymbol{\rho}$ update remains the same as that derived for the KL divergence case

$$\begin{aligned} {\boldsymbol{\rho}_*}^{t+1}(\beta, \boldsymbol{w}^t) &= \mathrm{Argmin}_{\boldsymbol{\rho}}\, \tilde{F}(\boldsymbol{w}^t, \boldsymbol{\rho}) + \frac{1-\beta}{\beta} D_\phi\big(\boldsymbol{w}^t \big|\big| \boldsymbol{\rho}\big) \\ &= \beta\left(\alpha \sum_i \boldsymbol{w}_i^t \boldsymbol{t}_i + (1-\alpha)\boldsymbol{s}\right) + (1-\beta)\boldsymbol{w}^t. \end{aligned} \tag{5.18}$$

The $\boldsymbol{w}$ updates are obtained as

$$\begin{aligned} \boldsymbol{w}^{t+1} &= \mathrm{Argmin}_{\boldsymbol{w}}\, \tilde{F}(\boldsymbol{w}^t, \boldsymbol{\rho}) + \frac{1-\beta}{\beta} D_\phi\big(\boldsymbol{w}^t \big|\big| \boldsymbol{\rho}\big) \\ &(\nabla\phi)^{-1}\left(\nabla\phi(\boldsymbol{\rho}^{t+1}) - \frac{\beta}{1-\beta} D_\phi\big(\boldsymbol{t}_i \big|\big| \boldsymbol{\rho}^{t+1}\big) - \lambda\right) \end{aligned} \tag{5.19}$$

where λ is the Lagrange multiplier enforcing the sum to 1 constraint. Comparing equation (5.19) with lemma (2) we can see that the 5.19 is the Bregman projection of $\boldsymbol{\rho}$ on a hyperplane whose normal direction is the vector $\overrightarrow{D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho})} - \lambda$. This has been shown for the KL divergence in figure 5.3.

From continuous differentiability of (5.17) and the fact that the alternate minimizers (5.18) and (5.19) are uniquely achieved it follows that (5.18 and 5.19) converges to a 0-gradient point of (5.17), which is weaker than what the Proposition 5 requires. What can we claim about these updates ? To make a quantitative claim, consider the function

$$J(\boldsymbol{\rho}) = \inf_{\boldsymbol{w}} \Big\langle \boldsymbol{w}, D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}) \Big\rangle + cD_\phi(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}) \leq \Big\langle \boldsymbol{w}, D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}) \Big\rangle + cD_\phi(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}) \quad (5.20)$$

that will be used as a surrogate.

Proposition 6. *Let* $\operatorname{dom}\phi(\cdot)$ *be bounded and* $\phi(\cdot)$ *be* s *strongly convex. Then iteration of updates (5.18, 5.19) with* $c_t = \frac{1-\beta_t}{\beta_t} \to \infty$ *converges to a* 0*-gradient point of surrogate* $J(\boldsymbol{\rho})$ *that is a lower bound of* (5.17) *that can be made arbitrarily tight and*

$$0 \leq \tilde{F}(\boldsymbol{\rho}, \boldsymbol{\rho}) - J(\boldsymbol{\rho}) = D_\phi(\boldsymbol{\rho} \Big|\Big| (\nabla\phi)^{-1} \left(\nabla\phi(\boldsymbol{\rho}) - \frac{\overrightarrow{D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho})}}{c} \right)) \leq \frac{1}{sc} \| \operatorname{dom}\phi \|^2.$$

Proof. Let us introduce a shorthand $\boldsymbol{d} = \overrightarrow{D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho})}$, then

$$
\begin{aligned}
&\tilde{F}(\boldsymbol{\rho}, \boldsymbol{\rho}) - J(\boldsymbol{\rho}) \\
&= \sum \rho_i D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}) - \inf_{\boldsymbol{w}} \left[\sum_i w_i D_\phi(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}) + c D_\phi(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}) \right] \\
&= \langle \boldsymbol{\rho}, \boldsymbol{d} \rangle + \sup_{\boldsymbol{w}} \left[\langle \boldsymbol{w}, -\boldsymbol{d} \rangle - c D_\phi(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}) \right] \\
&= \langle \boldsymbol{\rho}, \boldsymbol{d} \rangle + \sup_{\boldsymbol{w}} \left[\langle \boldsymbol{w}, -\boldsymbol{d} \rangle - c\phi(\boldsymbol{w}) + c\phi(\boldsymbol{\rho}) - c \langle \boldsymbol{\rho} - \boldsymbol{w}, \nabla\phi(\boldsymbol{\rho}) \rangle \right] \\
&= \langle \boldsymbol{\rho}, \boldsymbol{d} \rangle + c\phi^* \left(\frac{c\nabla\phi(\boldsymbol{\rho}) - \boldsymbol{d}}{c} \right) + c\phi(\boldsymbol{\rho}) - c \left\langle \boldsymbol{\rho}, \nabla\phi(\boldsymbol{\rho}) - \frac{\boldsymbol{d}}{c} \right\rangle - \langle \boldsymbol{\rho}, \boldsymbol{d} \rangle \\
&= c D_\phi(\boldsymbol{\rho} \Big|\Big| (\nabla\phi)^{-1} \left(\nabla\phi(\boldsymbol{\rho}) - \frac{\boldsymbol{d}}{c} \right)) \\
&= c D_\psi(\nabla\phi(\boldsymbol{\rho}) - \frac{\boldsymbol{d}}{c} \Big|\Big| \nabla\phi(\boldsymbol{\rho})) \leq c \frac{1}{s} \left|\left| \frac{\boldsymbol{d}}{c} \right|\right|^2 .
\end{aligned}
$$

Now update (5.19) can be recognized as tightening the bound (5.20) on $J(\boldsymbol{\rho})$ and update (5.18) as minimizing over the tightened bound, thereby ensuring convergence to 0-gradient point of $J(\boldsymbol{\rho})$. □

We now present results on the proximity of the solution to satisfying $\boldsymbol{w} = \boldsymbol{\rho}$ for a finite β, much analogous to section 5.2.2, but first recall a few preparatory relations.

Lemma 20. *The following three point property*

$$
D_\phi(\boldsymbol{x} \Big|\Big| \boldsymbol{y}) - D_\phi(\boldsymbol{x} \Big|\Big| \boldsymbol{z}) = D_\phi(\boldsymbol{z} \Big|\Big| \boldsymbol{y}) + \left\langle \overrightarrow{x_i - z_i}, \overrightarrow{\nabla\phi(y_i) - \nabla\phi(z_i)} \right\rangle
$$

holds for Bregman divergences.

Proof. Direct substitution of the definition of Bregman divergence yields the result. □

Lemma 21. *A Bregman divergence defined by a twice differentiable convex function $\phi(\cdot)$ that has a modulus of strong convexity s and whose gradient $\nabla\phi(\cdot)$ has a Lipschitz constant*

L can be bounded above and below as follows:

$$\frac{s}{2}\langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{x}-\boldsymbol{y}\rangle \leq D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big) \leq \frac{L}{2}\langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{x}-\boldsymbol{y}\rangle$$

Proof. For some $0<\alpha<1$ and $\boldsymbol{\chi}=\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}$ we have, by intermediate value theorem that $D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big)=\phi(\boldsymbol{x})-\phi(\boldsymbol{y})-\langle \boldsymbol{x}-\boldsymbol{y}, \nabla\phi(\boldsymbol{y})\rangle = \frac{1}{2}\left\langle \boldsymbol{x}-\boldsymbol{y}, \nabla^2\phi(\boldsymbol{\chi})(\boldsymbol{x}-\boldsymbol{y})\right\rangle$. Lipschitz constant L upper bounds the matrix norm of $\nabla^2\phi(\boldsymbol{\chi})$ whereas the modulus of strong convexity s lower-bounds the matrix norm, obtaining the proof. □

In general it will not be possible to specify β for which the updates converge to a solution that respect the equality constraint exactly. However, similar to Section 5.2.2 we can under specific conditions give an apriori bound on the value of β for which the constraints are satisfied to any arbitrary but finite degree of proximity.

Proposition 7. *In order to have the optimum $\tilde{\boldsymbol{\rho}}$ of the problem* (5.17) *satisfy the relation $\frac{\|\boldsymbol{\rho}_*-\tilde{\boldsymbol{\rho}}\|}{\|\boldsymbol{\rho}^*-\tilde{\boldsymbol{\rho}}\|} \geq \frac{\delta}{1-\delta}$ for any $\delta\in(0,1)$, it is sufficient that $\frac{1-\beta}{\beta} \geq L(\frac{1}{s}+\frac{\delta}{1-\delta})$.*

Proof. In order that there are no local minima between $\boldsymbol{\rho}^*$ and a arbitrary point $\tilde{\boldsymbol{\rho}}$ we require the following inequality to hold: (see figure 5.4)

$$\begin{aligned}\frac{1-\beta}{\beta}D_\phi\big(\boldsymbol{\rho}^*\Big|\Big|\tilde{\boldsymbol{\rho}}\big) &\geq D_\phi\big(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*)\Big|\Big|\boldsymbol{\rho}^*\big) - D_\phi\big(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*)\Big|\Big|\tilde{\boldsymbol{\rho}}\big)\\ &\overset{(a)}{=} D_\phi\big(\tilde{\boldsymbol{\rho}}\Big|\Big|\boldsymbol{\rho}^*\big) + \left\langle \overrightarrow{\rho_*-\tilde{\rho}_i}, \overrightarrow{\nabla\phi(\rho_i^*)-\nabla\phi(\tilde{\rho}_i)}\right\rangle.\end{aligned}$$

The equality (a) is obtained from lemma (20). By dividing both sides by $D_\phi\big(\boldsymbol{\rho}^*\Big|\Big|\tilde{\boldsymbol{\rho}}\big)$ we obtain the equivalent condition

$$\frac{1-\beta}{\beta} \geq \frac{D_\phi\big(\tilde{\boldsymbol{\rho}}\Big|\Big|\boldsymbol{\rho}^*\big)}{D_\phi\big(\boldsymbol{\rho}^*\Big|\Big|\tilde{\boldsymbol{\rho}}\big)} + \frac{\left\langle \overrightarrow{\rho_*-\tilde{\rho}_i}, \overrightarrow{\nabla\phi(\rho_i^*)-\nabla\phi(\tilde{\rho}_i)}\right\rangle}{D_\phi\big(\boldsymbol{\rho}^*\Big|\Big|\tilde{\boldsymbol{\rho}}\big)}.$$

We prove the result by upper bounding the quantity on the right hand side of the previous inequality. Substituting the lower and upper bounds obtained in lemma (21) we

	Penalty	Saddle Point
Convexity	Not Convex	Convex
$\boldsymbol{\rho}$ Update	Closed form	Closed form
$\boldsymbol{w}$ Update	Closed form or Numerical optimization	Closed form
Penalty Weight	Requires unbounded growth	Closed form
Numerical Stability*	Maybe unstable	Stable
Number of Iterations	Undetermined	Logarithmic

Table 5.1: A comparison of the penalty method and the saddle point based methods of consensus ranking. (*) This is an empirical observation and not a claim based on error sensitivity analysis. The tendency of the penalty terms to grow without bound in the penalty based method makes their updates numerically unstable.

can upper bound the first term by $\frac{L}{s}$. For the second term we invoke Cauchy -Schwarz inequality to yield:

$$\begin{aligned}\frac{\left\langle \overrightarrow{\rho_* - \tilde{\rho}_i}, \overrightarrow{\nabla\phi(\rho_i^*) - \nabla\phi(\tilde{\rho}_i)} \right\rangle}{D_\phi\left(\boldsymbol{\rho}^* \middle|\middle| \tilde{\boldsymbol{\rho}}\right)} &\leq \frac{\|\rho_* - \tilde{\rho}\| \, \|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\| L}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|^2} \\ &\leq L\left(\frac{\|\rho_* - \tilde{\rho}\|}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|}\right) \leq L\frac{\delta}{1-\delta}\end{aligned}$$

which completes the proof. □

5.3.2 Bregman Informatic Pessimistic Consensus and The Pagerank Game

Here we revisit formulation (5.5) in the context of Bregman divergences. Using properties of Bregman divergence we can simplify

$$\boldsymbol{w} = \text{Argmin}_{\boldsymbol{r}} \min_{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle$$

as $\boldsymbol{w} = T^* w$ where T^* is a matrix with columns $\boldsymbol{t}_i^*$ given by $\boldsymbol{t}_i^* = \min_{\boldsymbol{r}} \underset{\{\boldsymbol{t}_i \in \boldsymbol{T}_i\}}{\text{Argmin}} \langle \boldsymbol{w}, D(\boldsymbol{t}_i, \boldsymbol{r}) \rangle$. Now we will show that how one can solve (5.5) by dropping the fixed point condition by a reduction to a saddle point problems that does not have the fixed point constraint.

Proof. With specialization to Bregman divergence and the notational simplification one

obtains:

$$\max_{\boldsymbol{w}=T^*w,\langle \mathbf{1},\boldsymbol{w}\rangle=1} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| \boldsymbol{w}\big)\Big\rangle \tag{5.21}$$

$$\stackrel{a}{=} \max_{\boldsymbol{w}=T^*w} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| T^*\boldsymbol{w}\big)\Big\rangle + D_\phi\big(T^*\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big) - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big)$$

$$\stackrel{b}{=} \max_{\boldsymbol{w}=T^*w} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| \boldsymbol{r}^*\big)\Big\rangle + D_\phi\big(\boldsymbol{r}^* \Big|\Big| \boldsymbol{r}\big) - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big)$$

$$\stackrel{c}{=} \max_{\boldsymbol{w}=T^*w} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| \boldsymbol{r}\big)\Big\rangle - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big) \tag{5.22}$$

$$\stackrel{d}{=} \max_{\boldsymbol{w}} \min_{\boldsymbol{r}} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| \boldsymbol{r}\big)\Big\rangle - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big) \tag{5.23}$$

Equality (a) substitutes $\boldsymbol{w} = T^*w$ and adds and subtracts $cD_\phi\big(\boldsymbol{w}\big|\big|\boldsymbol{r}\big)$. Equality (b) follows from definition $\boldsymbol{r}^* = T\boldsymbol{w}^*$, see (5.2). Equality (c) follows from the property of Bregman divergence (C.8) (see Appendix). Note that in (d) the fixed point condition on $\boldsymbol{w}$ has been dropped. To see (d) note that, ignoring constants, (5.23) is equivalent to $\max_{\boldsymbol{w}} \min_{\boldsymbol{s}\in\operatorname{dom}\phi^*} \langle \boldsymbol{s}, T^*\boldsymbol{w} - \boldsymbol{w}\rangle$ which is unbounded unless $T^*\boldsymbol{w} = \boldsymbol{w}$ because $\operatorname{dom}\phi^*$ is unbounded by construction. We ensure boundedness by construction, by choosing the columns of T^* such that $\mathbf{1}^\dagger T^* = \mathbf{1}^\dagger$ ensuring that 1 is an eigenvalue. Note that the vector $\boldsymbol{s}$ acts as Lagrange multipliers for the fixed point condition, except that it is non-linearly related to $\boldsymbol{r}$ as $\nabla\phi(\boldsymbol{r}) = \boldsymbol{s}$. The same set of arguments can also be made to hold for $\sum_i w_i = c$ and choosing the columns of T such that $\mathbf{1}^\dagger T = c\mathbf{1}^\dagger$. □

For convenience we shall further assume and impose that the $\boldsymbol{r}$ component of the saddle point of (5.23) is located in the interior of $\operatorname{dom}\phi$. Note that this is consistent with the original pagerank algorithm because its *teleportation jumps* also imposes that the pagerank is obtained in the interior of the simplex. Recall that, for convex functions of the Legendre type, the norm of the gradient satisfies the following:

$$\lim_{\boldsymbol{r}\to\operatorname{bd}\phi} \|\nabla\phi(\boldsymbol{r})\| \to \infty$$

Therefore a convenient way to ensure that $\boldsymbol{r}$ remains in the interior is to add the constraint that $\|\nabla\phi(\boldsymbol{r})\| \leq \frac{1}{\epsilon}$ where ϵ is small positive number. With these changes we obtain:

$$\max_{\boldsymbol{w}} \min_{\boldsymbol{r} \mid \|\nabla\phi(\boldsymbol{r})\| \leq \frac{1}{\epsilon}} \Big\langle \boldsymbol{w}, D_\phi\big(\boldsymbol{t}_i^* \Big|\Big| \boldsymbol{r}\big) \Big\rangle - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{r}\big) \tag{5.24}$$

Before proceeding further, we quote the following Mini-Max theorem that will help us ensure the existence of and the convergence to a saddle point in our result 8.

Theorem 11. *(Rockafellar, 1996) page 393. Let $F(\cdot,\cdot)$ be a proper closed concave-convex function with domain $\mathcal{C} \times \mathcal{D}$. If either $\mathcal{C}$ or $\mathcal{D}$ is bounded its saddle point exists equivalently its minimax value equals its max-min value.*

In relation to the saddle point formulation of pagerank, we consider the objective function $G(\boldsymbol{w})$, defined variationally as

$$\begin{aligned} G(\boldsymbol{w}) &\triangleq \inf_{\boldsymbol{\rho} \mid \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} m(\boldsymbol{\rho}, \boldsymbol{w}) \\ &\triangleq \inf_{\boldsymbol{\rho} \mid \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \Big\langle \boldsymbol{w}, \overrightarrow{D_\phi\big(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}\big)} \Big\rangle - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}\big). \end{aligned} \tag{5.25}$$

The maximizer of $G(\boldsymbol{w})$ will be indicated as:

$$\boldsymbol{w}_\star = \operatorname{Argmax} G(\boldsymbol{w}).$$

Lemma 22. *The function $G(\boldsymbol{w}) \triangleq \inf_{\boldsymbol{\rho} \mid \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \Big\langle \boldsymbol{w}, \overrightarrow{D_\phi\big(\boldsymbol{t}_i \Big|\Big| \boldsymbol{\rho}\big)} \Big\rangle - D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}\big)$ is concave in $\boldsymbol{w}$ and strongly concave when $\phi(\cdot)$ is strongly convex.*

Proof. For any fixed value of $\boldsymbol{\rho}$ the cost function is concave in $\boldsymbol{w}$ because the first term is linear in $\boldsymbol{w}$ and a Bregman divergence $D_\phi\big(\boldsymbol{w} \Big|\Big| \boldsymbol{\rho}\big)$ is convex in its first argument $\boldsymbol{w}$. Since the cost function is a point-wise infimum of a family of concave costs, $G(\boldsymbol{w})$ is concave in $\boldsymbol{w}$. Strong concavity follows from the fact that every s-strongly convex function $\phi(\boldsymbol{x})$ is the sum of a convex function and $\frac{s}{2}\|\boldsymbol{x}\|^2$, therefore point-wise supremum of a family of $s-$

strongly convex functions is a summation of a convex function and $\frac{s}{2}\|\boldsymbol{x}\|^2$. $\square$

Given the local rank-score vectors $\boldsymbol{t}_i$ this leads us to propose the following consensus ranking problem, that is guaranteed to have a unique optimum

$$\sup_{\boldsymbol{w}} G(\boldsymbol{w}) = \sup_{\boldsymbol{w}} \inf_{\boldsymbol{\rho} \mid \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} m(\boldsymbol{\rho}, \boldsymbol{w}). \tag{5.26}$$

If equation (5.26) is to be optimized by using the $\sup$ and the $\inf$ operators, several key questions need to be resolved, among them are

- whether the Min-Max formulation is equivalent to the Max-Min formulation.
- whether it maintains uniqueness, and finally
- do these formulations replicate the pagerank solution.

We resolve all of these affirmatively. In order to do so, we shall consider another function

$$g(\boldsymbol{\rho}) \triangleq \sup_{\boldsymbol{w}} m(\boldsymbol{\rho}, \boldsymbol{w}). \tag{5.27}$$

Its minimizer will be indicated by

$$\boldsymbol{\rho}_\star = \mathrm{Argmin}_{\boldsymbol{\rho} \mid \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} g(\boldsymbol{\rho}).$$

Unlike $G(\boldsymbol{w})$, it is not straight forward to determine whether $g(\cdot)$ is convex. Even if we restrict ourselves to Bregman divergences that are jointly convex, it is not clear whether $g(\boldsymbol{\rho})$ is convex or concave, because $m(\cdot,\cdot)$ evaluated at a fixed $\boldsymbol{w}$ is a difference of convex functions. However for the case $\mathcal{W} \subset \{\boldsymbol{w} | \langle \boldsymbol{w}, \mathbf{1}\rangle = 1\}$ we can prove the following lemma.

Lemma 23. *The function $g(\boldsymbol{\rho})$ as defined in* (5.27) *with a set $\mathcal{W} \subset \{\boldsymbol{w} | \langle \boldsymbol{w}, \mathbf{1}\rangle = 1\}$ is a convex function in the variable $\nabla\phi(\boldsymbol{\rho})$ and differentiable when the maximizer over the set $\mathcal{W}$ in* (5.27) *is uniquely achieved.*

Proof. From equations (C.5) and (C.6) we obtain that for any $\boldsymbol{w} \in \mathcal{W}$ as defined, the function $m(\boldsymbol{\rho}, \boldsymbol{w}) = \left\langle \boldsymbol{w}, \overrightarrow{D_\phi\big(\boldsymbol{t}_i \big|\big| \boldsymbol{\rho}\big)} \right\rangle - D_\phi\big(\boldsymbol{w} \big|\big| \boldsymbol{\rho}\big)$ is a linear function of $\nabla\phi$. The function $g(\boldsymbol{\rho})$ is a point-wise supremum of a family of linear functions. Thus it is convex in $\nabla\phi$. For the proof of differentiability note that whenever the maximizer $\boldsymbol{w}^*$ of equation (5.27) is unique it defines an unique gradient for $g(\boldsymbol{\rho})$. □

In view of the special structure of the function $m(\boldsymbol{\rho}, \boldsymbol{w})$ we define a convex-concave function $M(\cdot, \cdot)$ as follows

$$M(\nabla\phi(\boldsymbol{\rho}), \boldsymbol{w}) \triangleq \begin{cases} m(\boldsymbol{\rho}, \boldsymbol{w}) & \text{if } \boldsymbol{w} \in \{\boldsymbol{w} |\, \langle \boldsymbol{w}, \mathbf{1} \rangle = 1\} \\ \infty & \text{otherwise} \end{cases}.$$

We are also able to verify the following claim:

Proposition 8. *Subject to the constraint* $\langle \mathbf{1}, \boldsymbol{w} \rangle = 1$, *choice* $\langle \mathbf{1}, \boldsymbol{t}_i \rangle = 1 \ \forall_i$, $\operatorname{dom}\phi \subset \{\boldsymbol{w} |\, \langle \mathbf{1}, \boldsymbol{w} \rangle = 1\}$ *for a convex function* ϕ *such that either* $\operatorname{dom}\phi$ *or* $\operatorname{dom}\phi^*$ *is bounded then the following mini-max (saddle point) equations are satisfied:*

$$\begin{aligned} \operatorname{Max}_{\boldsymbol{w}} G(\boldsymbol{w}) &= \operatorname{Max}_{\boldsymbol{w}} \operatorname{Min}_{\boldsymbol{\rho} \,|\, \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \left\langle \boldsymbol{w}, \overrightarrow{D_\phi\big(\boldsymbol{t}_i \big|\big| \boldsymbol{\rho}\big)} \right\rangle - D_\phi\big(\boldsymbol{w} \big|\big| \boldsymbol{\rho}\big) \\ &= \operatorname{Min}_{\boldsymbol{\rho} \,|\, \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \operatorname{Max}_{\boldsymbol{w}} \left\langle \boldsymbol{w}, \overrightarrow{D_\phi\big(\boldsymbol{t}_i \big|\big| \boldsymbol{\rho}\big)} \right\rangle - D_\phi\big(\boldsymbol{w} \big|\big| \boldsymbol{\rho}\big) \\ &\leq \phi^*(\phi(\vec{\boldsymbol{t}}_i)). \end{aligned} \tag{5.28}$$

The optimum $\boldsymbol{w}^*$ *satisfies the pagerank-stationarity condition* $T\boldsymbol{w}^* = \left[\boldsymbol{t}_1 \cdots \boldsymbol{t}_{|\mathcal{V}|}\right] \boldsymbol{w}^* = \boldsymbol{w}^*$.

Proof. Since $\langle \mathbf{1}, \boldsymbol{w} \rangle = 1$ we can invoke proposition (17), in particular equation (C.5) in the inner optimization over $\boldsymbol{\rho}$, as a result of which we obtain

$$\operatorname{Max}_{\boldsymbol{w}} G(\boldsymbol{w}) \leq \operatorname*{Max}_{\boldsymbol{w} \in \operatorname{dom}\phi} \operatorname*{Min}_{\boldsymbol{v} \in \operatorname{dom}\phi^*} -\phi(\boldsymbol{w}) + \left\langle \boldsymbol{w}, \phi(\vec{\boldsymbol{t}}_i) \right\rangle - [\boldsymbol{w}, -1]^\dagger \left[\boldsymbol{t}_1 \cdots \boldsymbol{t}_{|\mathcal{V}|}, \boldsymbol{w}\right] \boldsymbol{v}.$$

It is evident from the expression above and lemmata 23 and 22 that $m(\cdot,\cdot)$ is concave in $\boldsymbol{w}$ and linear in $\nabla\phi(\boldsymbol{\rho})$. If either $\operatorname{dom}\phi$ or $\operatorname{dom}\phi^*$ is bounded we can apply theorem 11 to switch the order of Min and Max and yet maintain equality.

The conditions $\langle \mathbf{1}, \boldsymbol{w}\rangle = 1$ and $\langle \mathbf{1}, \boldsymbol{t}_i\rangle = 1\ \forall_i$ ensure the existence of a vector $\check{\boldsymbol{w}}$ that satisfies $[\cdots \boldsymbol{t}_i \cdots]\check{\boldsymbol{w}} = \check{\boldsymbol{w}}$. We further assume that there exists a vector $\check{\boldsymbol{\rho}}$ such that $\check{\boldsymbol{w}}$ is the optimal vector obtained. The condition $\phi(\vec{\boldsymbol{t}}_i) \subset [\cdots \boldsymbol{t}_i \cdots] \operatorname{dom}\phi^*\ \forall \boldsymbol{t}_i$ ensures the existence of such a $\check{\boldsymbol{\rho}}$. With $\boldsymbol{w}$ fixed at $\check{\boldsymbol{w}}$ the corresponding optimal $\boldsymbol{v}$ is any vector in $\operatorname{dom}\phi^*$, certainly $\nabla\phi(\check{\boldsymbol{\rho}})$. Thus $\check{\boldsymbol{w}}, \check{\boldsymbol{\rho}})$ is a saddle point and $\check{\boldsymbol{w}}$ satisfies the pagerank-stationarity condition. □

Example 3. *Instantiating problem* (5.28) *for the KL divergence we observe that the result* (8) *applies because $\boldsymbol{w}$ lies in Δ which is a convex and compact set.*

Regret Bounded Algorithms applied to The pagerank Game

Proposition 8 reduces the objective function $G(\boldsymbol{w})$ and, under appropriate conditions, the cost function $g(\boldsymbol{\rho})$ to the two party game $\operatorname{Min}_{\boldsymbol{\rho}\ |\ \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \operatorname{Max}_{\boldsymbol{w}} m(\boldsymbol{\rho}, \boldsymbol{w})$. As a result of this reduction, any convex game solving algorithm may be applied to solve (5.26).

We choose to apply online "no-regret" algorithms to the saddle point problem in the setting of fictitious plays. Our choice is motivated by the balance between the simplicity of the individual updates and the convergence rate achieved Recall that ϕ is $1-$strongly convex by our choice, we show that for this case we can obtain a convergence rate of $\mathcal{O}(\frac{\log\tau}{\tau})$, where τ is the number of iterations.

Online Regret Minimization: We describe online regret minimization in the setting of maximizing concave functions because this is what we shall use, however such regret minimization algorithms can equally well be posed as minimizing convex functions.

At each time step t an online regret minimization algorithm has to commit to a prediction $\boldsymbol{w}_t \in \mathcal{R} \subset \operatorname{dom}\Gamma_t(\cdot)$, before the concave objective function $\Gamma_t(\cdot)$ is revealed. The subset $\mathcal{R}$ is convex and may be the entire domain. The instantaneous *regret* incurred

at stage t is defined as $\sup_{\boldsymbol{w}\in\mathcal{R}} \Gamma_t(\boldsymbol{w}) - \Gamma_t(\boldsymbol{w}_t)$ whereas the regret over the entire epoch $[1,\tau]$ is given by

$$R(\tau) = \sup_{\boldsymbol{w}} \sum_{t=1}^{\tau} \Gamma_t(\boldsymbol{w}) - \sum_{t=1}^{\tau} \Gamma_t(\boldsymbol{w}_t).$$

If for *any* sequence $\boldsymbol{w}_1 \cdots \boldsymbol{w}_\tau$ predicted by an online algorithm and for any $\Gamma_t(\cdot)$ drawn from some suitable subclass of $\mathcal{G}$ of concave functions and the following holds

$$\sup_{\boldsymbol{w}} \sum_{t=1}^{\tau} \Gamma_t(\boldsymbol{w}) - \sum_{t=1}^{\tau} \Gamma_t(\boldsymbol{w}_t) \leq C(\tau) \quad \forall \Gamma_t(\cdot) \in \mathcal{G}$$

then the algorithm is said to have a convergence rate of $C(\tau)$. The algorithm is called a "no-regret" algorithm if $C(\tau)$ is sub-linear in τ. Several classes of concave functions admit "no-regret" algorithms.

We now show how one may use such an algorithm for solving the saddle point problem (5.26). The updates to $\boldsymbol{w}$ will be obtained from a regret minimization algorithm targeting the instantaneous loss losses $\Gamma_t(\cdot) = M(\overline{\nabla\phi(\boldsymbol{\rho}_t)}, \cdot)$.

The $\boldsymbol{\rho}$ update, equivalently the $\nabla\phi(\boldsymbol{\rho})$ update will be greedy, point-wise optimal and for norms in the $\|\cdot\|_p$ family it will be obtained in closed form. In particular the $\nabla\phi(\boldsymbol{\rho})$ update becomes the *the norm duality mapping* and is unique if the norm $\|\cdot\|$ chosen is strictly convex.

Theorem 12. *() Consider a game defined by* $\mathrm{Min}_{\nabla\phi(\boldsymbol{\rho})} \mathrm{Max}_{\boldsymbol{w}} M(\nabla\phi(\boldsymbol{\rho}), \boldsymbol{w})$ *such that (i) the function* $M : (\nabla\phi(\boldsymbol{\rho}), \boldsymbol{w}) \mapsto \mathbb{R}$ *is convex in* $\nabla\phi(\boldsymbol{\rho})$ *and concave in* $\boldsymbol{w}$*; (ii) there is a "no-regret" online maximization algorithm for the sequence of optimization problems* $\mathrm{Max}_{\boldsymbol{w}} G_t(\boldsymbol{w})$ *where* $G_t(\boldsymbol{w}) \triangleq M(\nabla\phi(\boldsymbol{\rho}_t), \cdot)$ *with convergence rate* $C(\tau)$ *then*

$$\begin{aligned}\underset{\|\nabla\phi(\boldsymbol{\rho})\|\leq\frac{1}{\epsilon}}{\mathrm{Min}} \underset{\boldsymbol{w}}{\mathrm{Max}} M(\nabla\phi(\boldsymbol{\rho}), \boldsymbol{w}) &\leq M\left(\frac{1}{\tau}\sum_{t=1}^{\tau}\nabla\phi(\boldsymbol{\rho}_t), \frac{1}{\tau}\sum_{t=1}^{\tau}\boldsymbol{w}_t^*\right) \\ &\leq \underset{\boldsymbol{w}}{\mathrm{Max}} \underset{\|\nabla\phi(\boldsymbol{\rho})\|\leq\frac{1}{\epsilon}}{\mathrm{Min}} M(\nabla\phi(\boldsymbol{\rho}), \boldsymbol{w}) + \frac{C(\tau)}{\tau}\end{aligned}$$

Proof. Define $\boldsymbol{s}_t = \text{Argmin}_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} M(\boldsymbol{s}, \boldsymbol{w}_t)$, and $\bar{\boldsymbol{s}} \triangleq \frac{1}{\tau}(\sum_{t=1}^{\tau} \boldsymbol{s}_t)$. Let $\boldsymbol{s}_t = \nabla\phi\left(\boldsymbol{\rho}\right)_t$ and $\boldsymbol{w}_t*$ be obtained by a "no-regret" online maximization algorithm for the sequence of optimization problems $\text{Max}_{\boldsymbol{w}}\, M(\nabla\phi\left(\boldsymbol{\rho}_t\right), \cdot)$ with convergence rate $C(\tau)$ then

$$\underset{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}}{\text{Min}} \underset{\boldsymbol{w}}{\text{Max}} M(\boldsymbol{s}, \boldsymbol{w}) \leq \underset{\boldsymbol{w}}{\text{Max}} \frac{1}{\tau} M(\bar{\boldsymbol{s}}, \boldsymbol{w}) \overset{a}{\leq} \underset{\boldsymbol{w}}{\text{Max}} \frac{1}{\tau} \sum_{t=1}^{\tau} M(\boldsymbol{s}_t, \boldsymbol{w}) \tag{5.29}$$

$$\overset{b}{\leq} \frac{1}{\tau}\left(\sum_{t=1}^{\tau} M(\boldsymbol{s}_t, \boldsymbol{w}_t^*) + C(\tau)\right) \tag{5.30}$$

$$\overset{c}{\leq} \frac{1}{\tau} \sum_{t=1}^{\tau} \left(M(\boldsymbol{s}, \boldsymbol{w}_t^*) + C(\tau)\right) \text{ s.t. } \|\boldsymbol{s}\| \leq \frac{1}{\epsilon} \tag{5.31}$$

$$\overset{d}{\leq} \underset{\boldsymbol{s}}{\text{Min}}\, M\left(\boldsymbol{s}, \frac{1}{\tau} \sum_{t=1}^{\tau} \boldsymbol{w}_t^*\right) + \frac{C(\tau)}{\tau} \text{ s.t. } \|\boldsymbol{s}\| \leq \frac{1}{\epsilon} \tag{5.32}$$

$$\leq \underset{\boldsymbol{w}}{\text{Max}} \underset{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}}{\text{Min}}\, M(\boldsymbol{s}, \boldsymbol{w}) + \frac{C(\tau)}{\tau}. \tag{5.33}$$

Inequality (a) uses Jensen's inequality applied to the convexity of M in the first argument. Inequality (b) is obtained by using predictions $\boldsymbol{w}_t^*$ obtained by running a "no-regret" online maximization algorithm with rate of convergence $C(\tau)$ on the sequence of online objectives $M(\nabla\phi\left(\boldsymbol{\rho}_t\right), \cdot)$, (c) follows from point wise optimization of $\boldsymbol{s}_t$. Jensen inequality is applied on the second argument to obtain (d).

□

From (Shalev-Shwartz and Kakade, 2008) it follows that for strong convexity (concavity) we may choose an algorithm with $C(\tau) = \mathcal{O}(\log \tau)$.

Duality Mappings in Optimizing $\boldsymbol{\rho}$

The pagerank game solution algorithm proposed requires that at each step the following optimization problem be solved

$$\boldsymbol{s}_i = \text{Argmin}_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} M(\boldsymbol{s}, \boldsymbol{w}_i)$$

where $\boldsymbol{s}$ is related to $\boldsymbol{\rho}$ by the Legendre convex duality mapping $\boldsymbol{s}_i = \nabla\phi\left(\boldsymbol{\rho}\right)_i$. This mapping is crucial in turning a non-convex problem in $\boldsymbol{\rho}$ into a convex problem. Ignoring terms that do not depend on $\boldsymbol{s}$ we obtain

$$\boldsymbol{s}_i = \mathrm{Argmin}_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} \left\langle \boldsymbol{s}, \boldsymbol{w}_i - T\boldsymbol{w}_i \right\rangle.$$

Now note that this is exactly the duality mapping imposed by the norm $\|\cdot\|$ taken to be an ℓ_p norm. We thus also obtain that

$$\min_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} M(\boldsymbol{s}, \boldsymbol{w}_i) = \epsilon \|T\boldsymbol{w}_i - \boldsymbol{w}_i\|_*$$

where $\|\cdot\|$ is the dual norm of $\|\cdot\|$ this quantity can be looked upon as the deviation from satisfying the fixed point condition. The vector $\boldsymbol{s}_i$ is obtained in closed form and is unique if the norm $\|\cdot\|$ is strictly convex for example ℓ_p such that $0 < p < \infty$.

5.3.3 Recovering the Eigenvector Representation

In this short section we show that the saddle-point formulation is equivalent to the familiar eigenvector based formulation of pagerank. We show further that if the gradient of the dual, $\nabla\phi^*$ is available in closed form, as is the case for KL divergence, considerable algorithmic simplification can be obtained over the method proposed in Section 5.3.2. In what follows we shall use $\boldsymbol{s}$ for $\nabla\phi(\boldsymbol{\rho})$. First let us remove the constraint on $||\boldsymbol{s}||$ that we had imposed for numeric stability of the algorithm introduced in Section 5.3.2. From equation (5.28) we obtain the following by plugging in the definition of Bregman divergence, and Legendre-

Fenchel transform:

$$\begin{aligned}
&\min_{\boldsymbol{s}} \max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, \phi(\vec{\boldsymbol{t}}_i) \right\rangle - \phi(\boldsymbol{w}) - \langle \boldsymbol{s}, [T-I]\boldsymbol{w} \rangle \\
&\stackrel{a}{=} \min_{\boldsymbol{s}} \max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, \phi(\vec{\boldsymbol{t}}_i) + [I - T^{\dagger}]\boldsymbol{s}] \right\rangle - \phi(\boldsymbol{w}) \\
&= \min_{\boldsymbol{s}} \phi^* \left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^{\dagger}]\boldsymbol{s}] \right). \qquad (5.34)
\end{aligned}$$

The optimal $\boldsymbol{w}$ in sub-equation (a) is obtained as

$$\boldsymbol{w}_* = \nabla \phi^* \left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^{\dagger}]\boldsymbol{s}] \right). \qquad (5.35)$$

Equation (5.34) is a convex minimization problem in the variable $\boldsymbol{s}$.

$$\begin{aligned}
&[T-I]\nabla \phi^* \left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^{\dagger}]\boldsymbol{s}] \right) = 0 \\
&T\boldsymbol{w}_* = \boldsymbol{w}_* \quad \text{Using equation(5.40).} \qquad (5.36)
\end{aligned}$$

Note further that because of our assumptions of strong convexity on ϕ the cost function (5.34) las Lipschitz gradients and can therefore be minimized using accelerated gradient descent achieving a convergence rate of $\mathcal{O}(\frac{1}{\tau})$ in function value.

5.4 Consensus Ranking over Sets

In this section we finally address the problem of local and global consensus that we set out to accomplish in the introduction of the chapter, in particular in equation (5.4). The formulation (and consequently the algorithms) will be a direct generalization of what we used for the pagerank case in sections 5.3.1 and 5.3.2. The primary difference from the previously discussed pagerank scenario is that, instead of vectors $\boldsymbol{t}_i$ that represent the preference of vertex i we have to deal with a convex sets of uncertainty T_i associated with every vertex i. These sets represent the uncertainty over the set of weighted edges that emanate from the

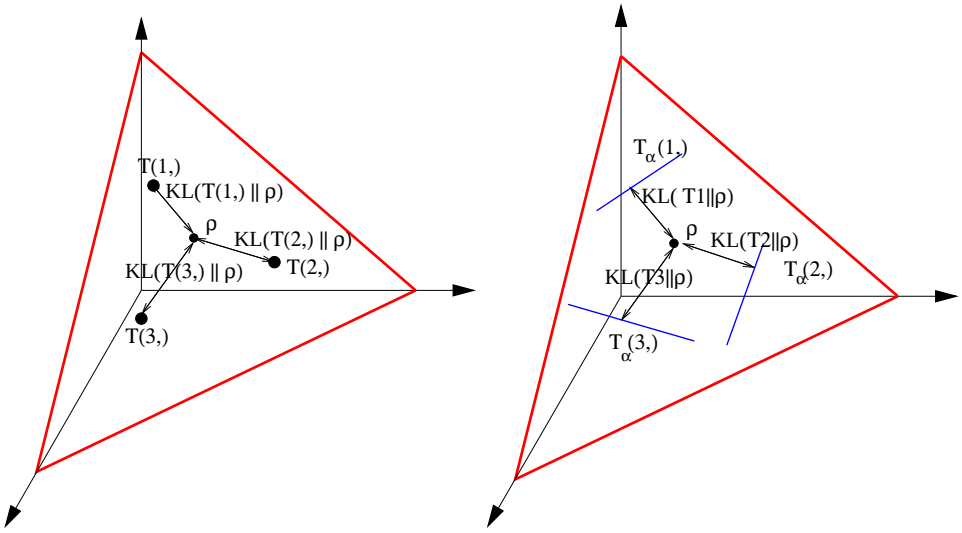


Figure 5.5: Left: A global consensus view of pagerank: The rank-score vector $\boldsymbol{\rho}$ is obtained as the minimizer of the weighted average KL divergences between the columns $T(i,)$ of the pagerank matrix and the rank-score vector $\boldsymbol{\rho}$. Right: A local-global consensus view of Brew rank: The rank-score vector $\boldsymbol{\rho}$ is obtained as the minimizer of the weighted average KL divergences between the convex sets in which the columns $T_{\alpha_i}(i,)$ of the effective pagerank matrix are allowed to lie and the rank-score vector $\boldsymbol{\rho}$. Additionally and crucially, the weights on the KL divergence terms have to be such that the The rank-score vector $\boldsymbol{\rho}$ is the stationary distribution of the effective pagerank matrix.

vertex. Each particular weighted edge set corresponds to a vector $\boldsymbol{t}_i \in \mathcal{T}(i,)$. The convex set $\mathcal{T}_i$ come about naturally in situations described in the introduction.

5.4.1 Bregman Informatic, Optimistic Consensus Over *Sets*

The consensus problem in this case is described by

$$\underset{T(i,\cdot)\in\mathcal{T}(i,\cdot)}{\text{Min}} \text{Min}_{\boldsymbol{\rho}} \sum w_i D_\phi\big(T(i,\cdot)\big|\big|\boldsymbol{\rho}\big) \quad \text{st} \quad \boldsymbol{w} = \boldsymbol{\rho}. \tag{5.37}$$

Since Bregman divergence is convex in the first argument and parameterization of $T_{\boldsymbol{\alpha}}$ is linear, the cost function has a global minimum for a fixed $\boldsymbol{w}$.

Note that the constraints are coupled because the consensus, in addition to being

$\boldsymbol{\rho}$ update: $\boldsymbol{\rho}^t = \sum_i \beta w_i^{t-1} T^{t-1}(i,\cdot) + (1-\beta)\boldsymbol{w}^{t-1}$

The cost function (5.38) is minimized with respect to $\boldsymbol{\rho}$ with the other parameters $\boldsymbol{w}$ and $T(i,\cdot)$ held fixed. Note that this equivalent to the condition of lemma 17 with weights on $T(i,\cdot)$ and $\boldsymbol{w}$ set to βw_i^{t-1} and $1-\beta$ respectively.

$T(i,\cdot)$ **update:** $T^{t+1}(i,\cdot) = \text{Argmin}_{\boldsymbol{p}}\, D_\phi\big(p\big|\big|\boldsymbol{\rho}^t\big)$ s.t. $\boldsymbol{p} \in T_{\alpha_i}(i,\cdot)$. This is a Bregman-projection computation of a distribution $\boldsymbol{\rho}^t$ on a linear set $T_{\alpha_i}(i,\cdot)$. The cost function totally decouples with respect to $T(i,\cdot)$..

$\boldsymbol{w}$ update:

$$w_i^{t+1} = \nabla\phi^{-1}\left(\nabla\phi(\boldsymbol{\rho}^t) - \boldsymbol{d}\right)$$
$$\boldsymbol{d} = [D_\phi\big(T_{\alpha_1}(1,\cdot)\big|\big|\boldsymbol{\rho}\big), \cdots D_\phi\big(T_{\alpha_i}(i,\cdot)\big|\big|\boldsymbol{\rho}\big)\cdots]^\dagger$$

With $T(i,\cdot)$ and $\boldsymbol{\rho}$ fixed, the $\boldsymbol{w}$ update is the well studied problem of finding a conjugate of a convex function (Rockafellar, 1996), in this case of $D_\phi\big(\boldsymbol{w}\big|\big|\boldsymbol{\rho}^t\big)$. The solution is obtained as the inverse of the conjugate. For $\text{KL}\left(\boldsymbol{w}\|\rho^t\right)$ is given in closed form by the sigmoid function.

Figure 5.6: Updates for Bregman Weighted (BreW) consensus Algorithm

close to the local recommendation sets, have to satisfy *stationarity*. This is our primary source of difficulty. An important question then is, *is there a way to compute the consensus in spite of the coupled nature of the stationarity constraints involved* by solving for a fixed $\boldsymbol{w}$ and then updating $\boldsymbol{w}$. The proposed algorithm works around this coupling by iteratively solving for a fixed $\boldsymbol{w}$ and then updating $\boldsymbol{w}$. As can be easily anticipated the formulation is the following:

$$\underset{T(i,\cdot)\in\mathcal{T}(i,\cdot)}{\text{Min}}\ \text{Min}_{\boldsymbol{\rho}}\,\text{Min}_{\boldsymbol{w}} \sum w_i D_\phi\big(T(i,\cdot)\big|\big|\boldsymbol{\rho}\big) + \frac{1-\beta}{\beta} D_\phi\big(\boldsymbol{w}\big|\big|\boldsymbol{\rho}\big). \tag{5.38}$$

The alternating minimization updates are shown in figure 5.6. A property of the expression (5.38) is that except for the requirement to apply Bregman Projection, the remaining updates are available in closed form (fig 5.6).

Proposition 9. *The minimum of expression* (5.38) *also minimizes* $\sum w_i D_\phi\big(T(i,\cdot)\big|\big|\boldsymbol{\rho}\big)$ *subject to the stationary constraints* $\sum w_i T(i,\cdot) = \boldsymbol{w} = \boldsymbol{\rho}$ *for some* $\beta = \beta^*$

Proof. The proof is in two parts, first proves that a minima exists that satisfies the pagerank stationarity condition and second that the fixed point will be reached by the updates.

Consider a scheme of coordinate descent updates where at iteration t $\boldsymbol{\rho}^t = \boldsymbol{w}$. The next update is given by $\boldsymbol{\rho} = \beta \sum_i w_i T(i,\cdot) + (1-\beta)\boldsymbol{w}$. For $\beta = \beta^*$ we have $\boldsymbol{\rho}^t = \boldsymbol{w} = \sum_i w_i T(i,\cdot)$, a stationary point. The cost function at this value is $\sum w_i D_\phi\big(T(i,\cdot)\big|\big|\boldsymbol{\rho}\big) + 0$. Since each coordinate descent update uniquely achieves a minimum of the bounded cost function, the iterations converge. □

5.4.2 Bregman Informatic, Pessimistic Consensus Over *Sets*

We propose two algorithms for Bregman informatic, pessimistic consensus over sets, (i) double loop BLend and (ii) single loop Blend. Both are very similar to the algorithm proposed in Section 5.3.2 for solving the equivalent problem over vectors. The difference from the algorithm proposed for vectors is that in addition to pointwise minimization of $\boldsymbol{s}_i$ one also optimizes over the choice of $T^{t+1}(i,\cdot)$.

For double loop BLend, one chooses $T^{t+1}(i,\cdot) = \operatorname{Argmin}_{\boldsymbol{p}i_i} D_\phi\big(\boldsymbol{p}_i\big|\big|\boldsymbol{\rho}^t\big)$ s.t. $\boldsymbol{p} \in T_{\alpha_i}(i,\cdot)\ \forall i$ jointly. Note that these variables are all decoupled except for coupling with $\boldsymbol{s}$. It can be shown that if ϕ is strongly convex with modulus of convexity 1 this joint minimization problem is also jointly convex. Thus one can optimize $\boldsymbol{s}_i$ and $T^{t+1}(i,\cdot)$ in an alternating minimization fashion till convergence and then update $\boldsymbol{w}$ and repeat. The proof of Theorem 12 and consequently the convergence rate remains unaffected by this change.

Double loop algorithms have to wait till the inner loop has converged and tend to be slow. As an alternative, one can have a single loop variant with a slightly worse constant

Initialize $\boldsymbol{\rho}^0$, $\boldsymbol{w}^0$, $\boldsymbol{t}_i$, set $t = 1$.

Choose τ the maximum number of iterations from convergence rate of the regret minimization algorithm employed in $\boldsymbol{w}$ **update**.

For $t = 1 \cdots \tau$ Repeat:

Repeat till convergence with t fixed

$\boldsymbol{\rho}$ **update:** $\boldsymbol{\rho}^{t+1} = \operatorname{Argmin}_{\boldsymbol{\rho} \| \|\nabla \phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \left\langle \boldsymbol{w}^t, \overrightarrow{D_\phi(\boldsymbol{t}_i \| \boldsymbol{\rho})} \right\rangle - D_\phi(\boldsymbol{w}^t \| \boldsymbol{\rho})$

$T(i, \cdot)$ **update:** $\boldsymbol{t}_i^{t+1} = \operatorname{Argmin}_{\boldsymbol{p}} D_\phi(p \| \boldsymbol{\rho}^{t+1})$ s.t. $\boldsymbol{p} \in T_{\alpha_i}(i, \cdot)$. The cost function totally decouples with respect to $T(i, \cdot)$.

$\boldsymbol{w}$ **update:** $\boldsymbol{w}^{t+1}$ Obtained from an online regret minimization algorithm for the sequence of optimization problems $\max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, D_\phi(\boldsymbol{t}_i \| \boldsymbol{\rho}^{t+1}) \right\rangle - D_\phi(\boldsymbol{w} \| \boldsymbol{\rho}^{t+1})$

Return $\frac{1}{\tau} \sum_{t=1}^{\tau} \boldsymbol{w}^t$.

Figure 5.7: Updates for double loop Bregman-Legendre saddle point (BLend) consensus ranking algorithm

of $\frac{2C(\tau)}{\tau}$ as shown by the following set of inequalities

$$\begin{aligned}
\operatorname*{Min}_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} \operatorname*{Max}_{\boldsymbol{w}} M((\boldsymbol{s}, T), \boldsymbol{w}) &\leq \operatorname*{Max}_{\boldsymbol{w}} \frac{1}{\tau} M(\overline{(\boldsymbol{s}, T)}, \boldsymbol{w}) \overset{a}{\leq} \operatorname*{Max}_{\boldsymbol{w}} \frac{1}{\tau} \sum_{t=1}^{\tau} M((\boldsymbol{s}_i, T_i), \boldsymbol{w}) \\
&\overset{b}{\leq} \frac{1}{\tau} \left(\sum_{t=1}^{\tau} M((\boldsymbol{s}_i, T_i), \boldsymbol{w}_i^*) + C(\tau) \right) \\
&\overset{c}{\leq} \frac{1}{\tau} \sum_{t=1}^{\tau} \left(M((\boldsymbol{s}, T_i), \boldsymbol{w}_i^*) + C(\tau) \right) \text{ s.t. } \|\boldsymbol{s}\| \leq \frac{1}{\epsilon} \\
&\overset{d}{\leq} \operatorname*{Min}_{\boldsymbol{s}} M \left((\boldsymbol{s}, T), \frac{1}{\tau} \sum_{i=1}^{\tau} \boldsymbol{w}_i^* \right) + \frac{2C(\tau)}{\tau} \text{ s.t. } \|\boldsymbol{s}\| \leq \frac{1}{\epsilon} \\
&\leq \operatorname*{Max}_{\boldsymbol{w}} \operatorname*{Min}_{\|\boldsymbol{s}\| \leq \frac{1}{\epsilon}} M(\boldsymbol{s}, \boldsymbol{w}) + \frac{2C(\tau)}{\tau}.
\end{aligned}$$

that are the same as (5.29) except for (d) which follows as a result of applying online regret minimization (with the same rate as that of the regret minimization applied to $\boldsymbol{w}$) on T. In the single loop variant the variables $\{\boldsymbol{t}_i\}$ are obtained by an online regret minimization

algorithm applied to the cost function sequence $\left\langle \boldsymbol{w}_t, \phi(\vec{\boldsymbol{t}}) \right\rangle - \phi(\boldsymbol{w}_t) - \langle \boldsymbol{s}, T\boldsymbol{w}_t - \boldsymbol{w}_t \rangle$, thereby eliminating the inner loop.

Initialize $\boldsymbol{\rho}^0$, $\boldsymbol{w}^0$, $\boldsymbol{t}_i$, set $t = 1$.

$\boldsymbol{s}^t = \nabla\phi(\boldsymbol{\rho}^t)\ \forall t$ and $T = [\boldsymbol{t}_1 \cdots]$.

Choose τ the maximum number of iterations from convergence rate of the regret minimization algorithm.

For $t = 1 \cdots \tau$ Repeat:

$\boldsymbol{\rho}$ **update:** $\boldsymbol{\rho}^{t+1} = \mathrm{Argmin}_{\boldsymbol{\rho} \| \|\nabla\phi(\boldsymbol{\rho})\| \leq \frac{1}{\epsilon}} \left\langle \boldsymbol{w}^t, \overrightarrow{D_\phi\left(\boldsymbol{t}_i \middle\| \boldsymbol{\rho}\right)} \right\rangle - D_\phi\left(\boldsymbol{w}^t \middle\| \boldsymbol{\rho}\right)$

$T(i, \cdot)$ **update:** $\boldsymbol{t}_i^{t+1}$ Obtained from an online regret minimization algorithm for the sequence of optimization problems $\min_{\{\boldsymbol{t}_i\}} \left\langle \boldsymbol{w}_t, \phi(\vec{\boldsymbol{t}}) \right\rangle - \phi(\boldsymbol{w}_t) - \left\langle \boldsymbol{s}^{t+1}, T\boldsymbol{w}_t - \boldsymbol{w}_t \right\rangle$.

$\boldsymbol{w}$ **update:** $\boldsymbol{w}^{t+1}$ Obtained from an online regret minimization algorithm for the sequence of optimization problems $\max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, D_\phi\left(\boldsymbol{t}_i \middle\| \boldsymbol{\rho}^{t+1}\right) \right\rangle - D_\phi\left(\boldsymbol{w} \middle\| \boldsymbol{\rho}^{t+1}\right)$

Return $\frac{1}{\tau}\sum_{t=1}^{\tau} \boldsymbol{w}^t$.

Figure 5.8: Updates for single loop Bregman-Legendre saddle point (BLend) consensus ranking algorithm

5.4.3 Using an Eigensolver

Similar to Section 5.3.3 the consensus algorithm can be reduced to eigenvector based updates. One may proceed exactly as (5.34) by eliminating $\boldsymbol{w}$ in closed form yielding a convex minimization problem in $\boldsymbol{s}$ and $\boldsymbol{t}_i$'.s

$$\begin{aligned}
&\min_{\boldsymbol{s}} \min_{\boldsymbol{t}_i \in T_{\alpha_i}(i,)} \max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, \phi(\vec{\boldsymbol{t}}_i) \right\rangle - \phi(\boldsymbol{w}) - \langle \boldsymbol{s}, [T - I]\boldsymbol{w} \rangle \\
&\stackrel{a}{=} \min_{\boldsymbol{s}} \min_{\boldsymbol{t}_i \in T_{\alpha_i}(i,)} \max_{\boldsymbol{w}} \left\langle \boldsymbol{w}, \phi(\vec{\boldsymbol{t}}_i) + [I - T^\dagger]\boldsymbol{s} \right\rangle - \phi(\boldsymbol{w}) \\
&= \min_{\boldsymbol{s}} \min_{\boldsymbol{t}_i \in T_{\alpha_i}(i,)} \phi^*\left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^\dagger]\boldsymbol{s}\right). \qquad (5.39)
\end{aligned}$$

The optimal $\boldsymbol{w}$ in sub-equation (a) is obtained as

$$\boldsymbol{w}_* = \nabla\phi^* \left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^\dagger]\boldsymbol{s}]\right). \tag{5.40}$$

The convex minimization problem can then be solved by alternating minimization. The $\boldsymbol{t}_i$ updates are Bregman projections, whereas the $\boldsymbol{s}$ is obtained via the eigenvector relation.

$$\begin{aligned} &[T - I]\nabla\phi^* \left(\phi(\vec{\boldsymbol{t}}_i) + [I - T^\dagger]\boldsymbol{s}]\right) = 0 \\ &T\boldsymbol{w}_* = \boldsymbol{w}_* \quad \text{Using equation(5.40).} \end{aligned} \tag{5.41}$$

5.5 Related Work and Discussion

The problem of rank aggregation has been studied both under a supervised Freund et al. (2003) as well as unsupervised scenario within a general and difficult combinatorial space of permutations with and without a probabilistic generative model Dwork et al. (2001), Lebanon and Lafferty (2002), Klementiev et al. (2009) and more recently in Qin et al. (2010). In this chapter consensus pagerank was posed as the solution of a constrained optimization problem posed in terms of Bregman divergences for which a convergent coordinate ascent, as well as online game playing algorithm was provided.

Chapter 6

Spam Resistant Ranking functions Using Convexity and Monotonicity

The ranking scheme of a search engine needs to be resistant to spam, a particularly sophisticated type of which is link-spam. Current countermeasures "de-spam" the corrupted webgraph by removing abusive pages identified by supervised learning. Since exhaustive detection and neutralization is infeasible, there is a need for ranking functions that can, on one hand, attenuate the effects of link-spam without supervision and on the other hand, counter spam more aggressively when supervision is available. A family of non-linear, iteratively defined functions is proposed that propagate "rank" and "trust" scores through the webgraph. It includes Pagerank as a special case and relies on non-linearity and convexity to provide the spam resistance. The main contributions of this chapter are (i) the proof of convergence and uniqueness of the iterates, and (ii) empirical comparison with Pagerank and other established anti-spam rankings on a part of the real webgraph with 13 million edges. The well known linear algebraic proof of convergence of Pagerank do not apply to this non-linear family. Hence different techniques are adopted and adapted. It is verified experimentally that spam resistance of the proposed *unsupervised* variant is comparable to the *supervised* state-of-the-art anti-spam techniques of Trust rank Gyongyi et al. (2004),

AntiTrustrank Krishnan and Raj (2006) and Demotedrank Wu et al. (2006). On the other hand when labels are available the proposed scheme can improve performance over the established state of the art. Though non-linearity is critical to the enhanced performance, it is not universally beneficial. It is experimentally shown and logical argued that best results are obtained by non-linear update for the propagation of "rank-score" but linear updates for the propagation of "trust scores".

Given a query, a search engine returns a list of web pages, ranked according to a combination of their content and *topological* (link analytic, graph theoretic) quality. The topological quality, an example of which is Pagerank Brin and Page (1998), is customarily measured by a real number Kleinberg (1999a) also called "rank" or "score". It is not just the order induced by these rank-scores but also the rank scores themselves (say Pagerank) that are combined with other signals to determine the final ordered list presented to the user. Because of the importance of the ordering as well as the values of the scores, we evaluate both the quantities in our experiments. The ordering are compared by precision-recall curves and Spearman foot-rule distance, whereas the scores are compared by the cumulated score assigned to spam pages, the lower the better.

Incorporation of topological quality has been critical to the success of search engines because content based information retrieval (IR) scores have been relatively easy to spam. Pagerank, a popular and effective link analysis score, though harder to manipulate than an IR score, is not immune to link-spam Henzinger (2003). Often several low quality pages point to and hence direct sufficient rank mass towards the spammed page through what is known as a Sybil attack Douceur (2002). Our objective is to be more resistant to such and other attacks. While it is unrealistic to assume the proposed scheme will be inherently immune to all possible attack modes to emerge in the future, it can adapt to them provided examples of spam and non-spam are provided.

A key difference between the proposed and prevalent methods is that the proposed method can function without a set of spam pages pre-identified. It can however benefit from

identification if available. Its spam resistance is compared with Trustrank Gyongyi et al. (2004), Demotedrank Wu et al. (2006) and AntiTrustrank Krishnan and Raj (2006) these are Pagerank like iterative algorithms that use supervisory labels. Guarantees of convergence and uniqueness are provided for the proposed iterative method. Without such a guarantee, an iterative ranking scheme is of questionable merit because there is no principled way to argue that the ranks at some *arbitrary* iteration number or initialization will possess the desired qualities. Without these properties, one is simply hiding the task of ranking under the tasks of (i) choosing a good initial condition and (ii) the choice of the final iteration.

A few words about notation: matrices are denoted by upper case letters such as A, whereas vectors by lower case letters in bold, such as $\boldsymbol{r}$. $\mathbf{1}$ denotes a column vector of all 1s. Transpose and inverse of a matrix A is denoted by $A^\dagger$ and A^{-1} respectively. Script fonts are used for sets, $\mathcal{V}$ is used to represent vertices of a graph, and $\mathcal{E}$ its edges. The in and out degree of a vertex v_i is denoted by $d_{in}(i)$ and $d_{out}(i)$ respectively. Functions mapping $\mathbb{R}^n \mapsto \mathbb{R}^n$ are denoted by upper case letters. $Eig(\cdot)$ denotes the principal eigenvector of its argument, a matrix.

6.1 Pagerank and its Relatives

Link analysis based techniques rank order nodes of a graph $G(\mathcal{V}, \mathcal{E})$ based on their topological properties. For example in the Pagerank model Brin and Page (1998), each page (a vertex of the webgraph joined by hyperlinks as edges) distributes its rank-score equally among its out-neighbors. The Pagerank of a page is the corresponding flow of rank-score at equilibrium. Hence it is the inverse out-degree weighted Pagerank of its in-neighbors. Pageranks can also be interpreted as the stationary distribution of a random-walk that picks an outlink to follow from the current page uniformly at random or resets to a random page on the web in a way described next (with probabilities $1 - \alpha$ and α respectively).

Let A be the adjacency matrix of the graph, D_{out} the diagonal matrix formed by the out-degrees, S a row stochastic source matrix usually taken to be $S = \frac{1}{N}(\mathbf{1} \times \mathbf{1}^\dagger)$, where $\mathbf{1}$

is a column vector of ones and N is the size of the vertex set. The transition matrix T of the walk can be expressed in terms of the out-degree normalized adjacency matrix $D_{out}^{-1}A$, and the random jump probability α as :$T = (1-\alpha) \times D_{out}^{-1} \times A + \alpha \times S$. Some vertices of the graph may have no outlinks in which case the D_{out}^{-1} has to be specially defined. This is the problem of "dangling links", the reader is referred to Brin and Page (1998) and Acharyya and Ghosh (2004) on how this can be dealt with. The Pagerank iteration converges to the primary eigenvector $\boldsymbol{\pi}$ of $T^{\dagger}$ or, equivalently, the steady-state probabilities of the Markov chain defined by it.

Trustrank is a supervised mechanism Gyongyi et al. (2004) to counter link-spam. Trust score is allowed to propagate out through the graph, much like Pagerank but from human verified "good", non-spam source pages. Trustrank propagates distrust from the spam pages in the same direction as that of the links, this can however be adversarially abused in the following way: since Trustrank believes in *"guilt by association"*, any page can be demoted by a spammer by pointing to it. This can be countered if "guilt" is propagated in a direction opposite to that of the hyperlinks. In this case a page gets demoted if the page itself points to a spam page, not if it is pointed to by one. Hence in our experiments distrust is taken to flow in a direction opposite to the links as is the case for similar approaches in Demotedrank Wu et al. (2006) and AntiTrustrank Krishnan and Raj (2006).

The formulation that is most similar in spirit to ours is Baeza-Yates et al. (2006). There the functional form of damping of the rank-score received by a page is generalized to include those that are non exponential in the path length whereas for Pagerank it is exponential. The stress in Baeza-Yates et al. (2006) is the nature of the decay and the different generalizations that are obtained and not on guarantees convergence, uniqueness or spam tolerance. The last three properties ignored in Baeza-Yates et al. (2006) are of vital importance.

6.2 Concave-Convex Rank

To understand the spam susceptibility of Pagerank, let us express the Pagerank r_i of page $v_i \in \mathcal{V}$ as the composition of two functions of the Pageranks $\{r_j\}_{j\in\mathcal{I}(i)}$ of its in-neighbors $\mathcal{I}(i)$ and their out degrees $\{d_{out}(j)_{j\in\mathcal{I}(i)}\}$ as: $r_i \propto g(f^i(\{r_j\},\{d_{out}(j)\}))\ s.t.\ j \in \mathcal{I}(i)$.

The function $g : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is identity for Pagerank algorithm and the function $f^i : {\mathbb{R}_+}^{2d_{in}(i)} \mapsto \mathbb{R}_+$ is a weighted sum, accumulating the ranks of the inlinking pages. The function f^i serves the purpose of accumulating the ranks into net input rank flow, and g maps the net rank flow into its rank score. The functions f^i for different i have the same functional form, the superscript i indicates that they operate on domains of different sizes depending on the neighbors of i. The choices of f^i and g in the Pagerank formulation entail a couple of spam susceptible properties: (i) g being identity, ensures that there is no *diminishing rate of return*. Lack of diminishing rate of return means that a link from a source increases the Pagerank of the recipient pages equally, irrespective of whether the recipient already has hundreds of links or just one. In other words the return obtained by virtue of receiving a link does not diminish. Secondly, (ii) because f^i is a sum, an inlink from a high quality page is worth as much as getting a 1000 links from low quality pages with Pagerank $1/1000$ths the value of the high quality one.

It is generally held to be true that the increment in the human perceived quality of a page diminishes with each link received, and a page with several poor inlinks is almost certainly worse than a page that receives few links but from high quality pages. For example a link from $www.yahoo.com$ could be equaled by thousands of links from worthless pages and it does not cost much to create such numerous dynamic pages on the web. Pagerank's teleportation property ensures all of them receives a certain low fraction of the web's total rank, all of which can be channelled into a spammed page to increase its rank.

We list two properties of g and f^i that would provide some spam-resistance to the ranking function. We make particular *choices*, based on simplicity of the over-all scheme and requirements of convergence of the ranks to a unique value. We do not claim any

theoretical sufficiency of the properties mentioned, but argue and experimentally show that the resulting ranks are more spam resistant than their peers. We list the properties desired of g first, it should be:

1. monotonically increasing: $g(x) > g(y), \forall\, x > y$.

   Between pages that have different net ranks flowing in, the page with more flow has higher quality. This is captured by the monotonicity property.

2. have diminishing rate of return: $g'(x) = \frac{\partial}{\partial x} g(x)$ is monotonically decreasing. This models the fact the increment in quality decays with the flow of rank. This also implies that the function $g()$ is concave.

   Of the limitless possibilities we *choose* to be conservative in letting this decay be polynomial as opposed to exponential, i.e. $\frac{\partial}{\partial x} g(x) = O(1/x^i)$ where $x, i > 0$. This leads us to functions of the form $\frac{x^{1/q}}{1/q} \quad q > 1$.

For the function f^i we desire that, between two pages with the same total rank flow, as measured by $\sum \frac{r_j}{d_{out}(j)}$, it allocate higher value for the cases where a few high ranked pages point to it as opposed to several low ranked ones. The functional requirements can be formalized by the following set of equations:

1. Existence of minima: $\forall \boldsymbol{x} \in \mathbb{R}^n \;\; f^i(\boldsymbol{x}) \geq f^i(\overline{\boldsymbol{x}}) \quad \overline{x}_k = \frac{\sum_{k \in \mathcal{I}(i)} x_k}{n}$, where $x_k = \frac{r_k}{d_{out}(k)}$.

   $\overline{\boldsymbol{x}}$ and $\boldsymbol{x}$ are equi-dimensional vectors with each component of $\overline{\boldsymbol{x}}$ equal to the average of the components of $\boldsymbol{x}$. Since $\sum x_k = \sum \overline{x_k}$, the property above favors few good inlinks over several mediocre ones.

2. Monotonically increasing: $f^i(\boldsymbol{x}) \geq f^j(\boldsymbol{y}) \,\forall\, \boldsymbol{x}, \boldsymbol{y} \geq 0$ and $\boldsymbol{y}$ is an extension of $\boldsymbol{x}$ formed by increasing its dimensionality by additional non-negative components to $\boldsymbol{x}$.

If we assume the permutation independent form $f^i(\boldsymbol{x}) = \sum_k f(x_k)$, it is sufficient for the properties above to hold that each $f(\cdot)$ is convex as shown below.

Lemma 24. *Given a convex function $f(\cdot)$, the function $f^i(\boldsymbol{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ such that $f^i(\boldsymbol{x}) = \sum_k^n f(x_k)$ satisfies the property $\forall \boldsymbol{x} \in \mathbb{R}^n \quad f^i(\boldsymbol{x}) \geq f^i(\overline{\boldsymbol{x}})$*

Proof. $f^i(\boldsymbol{x}) = n \times \sum_k^n 1/n f(x_k) \geq n f(\sum_k \frac{x_k}{n}) = f(\overline{\boldsymbol{x}})$ by Jensen's inequality. □

Convex functions have monotonically increasing derivatives. Out of the limitless possibilities of convex functions we take the less aggressive *choice* that its derivative exhibit only polynomial increase i.e. $\frac{\partial}{\partial x} f_k(x_k) = O(x_k^p)\ s.t.\ x_k, p > 0$. For the purpose of this chapter we make the choice that $f_k(x_k) = (\frac{x_k}{p})^p$, $p > 1$.

As an example and for reasons of simplicity of exposition we first choose matching indices p and q in functions f^i and g such that we have

$$g(f^i(\boldsymbol{x})) = p(f^i(\boldsymbol{x}))^{\frac{1}{p}} \left(\sum_j (x_j)^p \right)^{\frac{1}{p}} = ||\boldsymbol{x}||_p$$

where $||x||_p = [\sum |x_i|^p]^{1/p}$ is the L_p norm of the vector $\boldsymbol{x}$. [1] Tsaparas independently Tsaparas (2004) considered NORM() and MAX() in his thesis and proved their convergence. Here we show that the Concavo Convex ranks subsumes those results.

We state again that we are not championing the case for L_p norms for ranking, but use it as an instrument of exposition, and as a strong baseline for experiments and to motivate the functional form L_{pq} that we actually use, where p and q are different. Our experiments indicate that the NORM() family performs worse than the L_{pq} ranks. Also note that the NORM() family is equivalent to choosing the components of $f^i()$ to be x^p and g its inverse $x^{1/p}$. Thus under the transformation $\boldsymbol{r'} = \boldsymbol{r}^p$ the update $\boldsymbol{r'} = T^\dagger \boldsymbol{r'}$ is linear and the theory of eigenvectors of linear operators suffice as a proof of convergence. NORM()

[1] As an aside we demonstrate that this simple form covers the logarithmic function in the limiting condition. This is significant because the most commonly used function where rate of diminishing return is desired is the log function We show that as $k \to \infty$ the chosen g goes to log on the positive orthant

Proposition 10. $\forall_x \in \mathbb{R}_+ \quad \lim_{k\to\infty} \frac{x^{1/k}}{1/k} = \log(x)$

rank is a simple nonlinear transformation and normalization of the ranks *after the ordinary Pagerank iteration has converged*.

The important properties of uniqueness of the computed ranks and the convergence of the updates from any initialization will be proven shortly for any $p \leq q$. In fact our experiments show that setting q higher than p significantly benefits not only the speed of convergence but also the spam resistance. We conjecture this happens because the ranking function as a whole becomes concave, whereas norms are convex, thus devoid of the property of diminishing returns. We emphasize again that L_p is introduced to aid the description and to serve as a baseline for experiments, the method that we propose for actual use are the L_{pq} family with $q > p$, not the L_p family.

For $d_{out}(i)$ the out-degree of a page, $d_{in}(i)$ the in-degree of a page, and $\mathcal{I}(i)$ the in-neighbors of a page, the update equation for the ranks is obtained by substituting $\frac{r_j}{d_{out}(j)}$ for x_j. Some modification is necessary to take care of loops and absorbing vertices of the graph. Absorbing nodes are eliminated by adding "weak links" from all vertices to all other vertices. These links are called weak because they are designed to transmit only a small fraction of the rank. The algorithm, adjusted for the presence of absorbing nodes and generalized to have non-matching exponents is presented in figure 6.1. Owing to the similarity of the function used to L_p norms we call it the $L_{p,q}$ algorithm. We omit the parameter q whenever we assume that $p = q$. Let us consider the implications of our choice

Iterate $r_i^{t+1} =$

$$\left|\left| \alpha \left[\frac{r_j^t}{d_{out}(j)} \right]_{j \in \mathcal{I}(i)} + \frac{(1-\alpha)}{N} ||\boldsymbol{r}^t||_1 \right|\right|_p^{p/q} \triangleq U_i(\boldsymbol{r}^t)^a \qquad (6.1)$$

Normalize $\boldsymbol{r}^{t+1} = \frac{\boldsymbol{r}^{t+1}}{||\boldsymbol{r}^{t+1}||_1}$

[a] proofs remain valid if the weak links are taken out of the norm leading to a convex combinations of the weights due to strong and weak links.

Figure 6.1: $L_{p,q}$ Rank Algorithm

for the matching exponent case:

1. We obtain the Pagerank updates for $p = q = 1$
2. An interesting special case is when $p = q = \infty$, where the L_p rank is the maximum inflow from among the incoming edges. This is robust against spam because the rank of a page cannot now be manipulated by adding low quality spam links and is computationally cheap.
3. If all the in neighbors' ranks are incremented by q_j the L_p rank of the page is incremented by less than $||\boldsymbol{q}||p$.
4. If all the in-neighbors' ranks are scaled by β the L_{pq} rank of the page is also scaled by $\beta^{\frac{q}{p}}$.

The update in equation 6.1 in figure 6.1 can be looked upon as a function $U : \boldsymbol{x} \mapsto \boldsymbol{y} \quad \boldsymbol{x} \in \Delta^{N-1}$ $\boldsymbol{y} \in {\mathbb{R}_+}^N$ where N is the size of the vertex set of the graph G and Δ^{N-1} is a unit regular N dimensional simplex. Important considerations are the existence and uniqueness of the fixed points of $\boldsymbol{r}^{t+1} = \frac{1}{||U(\boldsymbol{r}^t)||_1} U(\boldsymbol{r}^t)$.

- Does the scheme have a fixed point ?
- Are the fixed points stable ?
- Is the fixed point unique ?
- Does any initialization converge to a fixed point ?
- What is the rate of convergence for the iteration ?

In the remaining part of this section we resolve these issues. The answer is yes for the first four but unresolved for the last. The results on fixed points that we will mention below have been derived from those stated for the case of homogeneous functions of degree 1 in the context of economics Robert.M.Solow and Paul.A.Samuelson (1953). For our application we extend the scope to homogeneous, and super-homogeneous functions of degree less than and equal to one. This is a very large family of functions for which we can give convergence

and uniqueness guarantees. Since the domain and range of U is the simplex Δ^N and hence a closed convex set and U is continuous, Brouwer's Fixed Point Theorem Rudin (1976) ensures the existence of a fixed point. For the current application in mind, it is desirable that the fixed point be unique. We investigate sufficient conditions for unique non-linear eigenvectors of the function $U : \boldsymbol{x} \mapsto \boldsymbol{y}; \quad \boldsymbol{x}, \boldsymbol{y} \in \Delta^N \subset {\mathbb{R}_+}^N$. A vector $\boldsymbol{x} \in \mathbb{R}^n$ is said to be greater than another vector $\boldsymbol{y} \in \mathbb{R}^n$, i.e. $\boldsymbol{x} \geq \boldsymbol{y}$ if $\forall_i x_i \geq y_i \text{ and } \boldsymbol{x} \neq \boldsymbol{y}$

Definition 1. *Function $F : \boldsymbol{x} \mapsto \boldsymbol{y},\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ is positively homogeneous, sub homogeneous or super homogeneous of degree α if $\forall_{c>1}\ F(c\boldsymbol{x}) = c^\alpha F(\boldsymbol{x})$, $F(c\boldsymbol{x}) \leq c^\alpha F(\boldsymbol{x})$, $F(c\boldsymbol{x}) \geq c^\alpha F(\boldsymbol{x})$ respectively and increasing if $\forall \boldsymbol{x} \geq \boldsymbol{y}\ F(\boldsymbol{x}) \geq F(\boldsymbol{y})$*

Lemma 25. *If an increasing function $U : \boldsymbol{x} \mapsto \boldsymbol{y}\ \boldsymbol{x}, \boldsymbol{y} \in {\mathbb{R}_+}^N$ is positively homogeneous of order $\alpha = 1$ then the eigenvalue associated with different eigenvectors is unique, furthermore if U is positively super homogeneous of degree of homogeneity $\alpha < 1$ then eigenvectors are unique.*

Proof. We prove the proposition for homogeneous function, extending it to super homogeneous functions is mostly matter of change of the symbol $\doteq$ to the corresponding inequality. We have $U(\boldsymbol{x}) \geq 0 \text{ and } \forall \boldsymbol{x} \geq \boldsymbol{y} \quad U(\boldsymbol{x}) \geq U(\boldsymbol{y})$ and $\forall c > 0 \quad U(c\boldsymbol{x}) \doteq c^\alpha U(\boldsymbol{x})$. Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two eigenvectors with the corresponding eigenvalues λ and κ. Let M be a scalar such that $\forall_i \frac{u_i}{M} < v_i$, such a number always exists. Therefore $\kappa \boldsymbol{v} = U(\boldsymbol{v}) \geq U(\frac{\boldsymbol{u}}{M}) \doteq \frac{1}{M^\alpha} U(\boldsymbol{u}) = \frac{\lambda}{M^\alpha} \boldsymbol{u}$ or, $\boldsymbol{v} \geq \left(\frac{\lambda}{\kappa}\right) \frac{1}{M^\alpha} \boldsymbol{u}$. By applying the relation above n times we obtain

$$\boldsymbol{v} \geq \left(\frac{\lambda}{\kappa}\right)^{\sum_{i=0}^{n-1} \alpha^i} \frac{1}{M^{\alpha^n}} \boldsymbol{u} = \begin{cases} \left(\frac{\lambda}{\kappa}\right)^{\frac{1}{1-\alpha}} \boldsymbol{u} & \text{if } \alpha < 1 \\ \left(\frac{\lambda}{\kappa}\right)^n \frac{\boldsymbol{u}}{M} & \text{if } \alpha = 1 \end{cases} \tag{6.2}$$

, implying $\lambda \geq \kappa$. Selecting another constant N such that $\forall_i \frac{v_i}{N} < u_i$ one can reverse the roles of $\boldsymbol{u}$ and $\boldsymbol{v}$ implying $\lambda \leq \kappa$, this can be true for $\alpha \leq 1$ only if $\kappa = \lambda$. Note for $\alpha < 1$ it also implies that the eigenvectors $\boldsymbol{u} = \boldsymbol{v}$. Equality of the eigenvectors is not obvious for $\alpha = 1$ this is established in lemma 26. □

Definition 2. *An increasing function $F(\boldsymbol{x}) \mapsto \boldsymbol{y},\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ is irreducible if there can be no permuted partition of its input $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{pmatrix}$ and $\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \end{pmatrix}$ for $F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$ such that $\boldsymbol{x_1} = \boldsymbol{y_1}$ and $\boldsymbol{x_2} \geq \boldsymbol{y_2}$ which has $F_1(\boldsymbol{x}) \leq F_1(\boldsymbol{y})$.*

Note that the above definition is a generalization of irreducibility of matrices to functions. With the above definition in place we can now lay down the condition for a unique eigenvector in the following theorem.

Lemma 26. *Given a positive increasing homogeneous function $U : \boldsymbol{x} \mapsto \boldsymbol{y} \quad \boldsymbol{x}, \boldsymbol{y} \in {\mathbb{R}_+}^N$ that is irreducible, the corresponding normalized function $\hat{U} : \boldsymbol{x} \mapsto \boldsymbol{y} \quad \boldsymbol{x}, \boldsymbol{y} \in \Delta^{N-1}$ has a unique eigenvector.*

Proof. Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two different eigenvectors of U with eigenvalue λ. Let M be a positive scalar such that $v_i \leq M u_i$, we permute and partition the function U and its input so that $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v_1} \\ \boldsymbol{v_2} \end{pmatrix}$ and $\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{pmatrix}$ and $\boldsymbol{v_1} =^{(a)} M\boldsymbol{u_1}$ and $\boldsymbol{v_2} < M\boldsymbol{u_2}$. Now consider a perturbation $\tilde{\boldsymbol{v}} = \begin{pmatrix} \boldsymbol{v_1} \\ \tilde{\boldsymbol{v_2}} \end{pmatrix}$ such that $\tilde{\boldsymbol{v_2}} = M\boldsymbol{u_2}$, i.e. $\tilde{\boldsymbol{v}} = M\boldsymbol{u}$. We have $U(\boldsymbol{v}) = \lambda \boldsymbol{v}$, since U is irreducible we have $U_1(\tilde{\boldsymbol{v}}) >^{(b)} U_1(\boldsymbol{v})$. LHS equals $U_1(M\boldsymbol{u}) = \lambda M \boldsymbol{u}_1$ and RHS equals $U_1(\boldsymbol{v}) = \lambda \boldsymbol{v}_1$. Equations (a) and (b) contradict hence $\hat{\boldsymbol{u}} = \hat{\boldsymbol{v}}$. □

Theorem 13. *If an increasing function $U : \boldsymbol{x} \mapsto \boldsymbol{y}\ \boldsymbol{x}, \boldsymbol{y} \in {\mathbb{R}_+}^N$ is irreducible and positively homogeneous of order $\alpha = 1$ or is positively super homogeneous of degree of homogeneity $\alpha < 1$ then eigenvector of $\frac{U}{||U||}$ is unique.*

Proof. follows from lemmas 25, 26. □

Corollary 6. *Given a graph $G(\mathcal{V}, \mathcal{E})$ the L_{pq} ranks defined by equation 6.1 has a unique fixed point.*

Proof. Since the update equations are a linear composition of norms they are positive homogeneous with degree 1 (or less than one for the non matching case of $q > p$) and monotonic

increasing. Since every vertex contributes positively to the rank of every other vertex the function is irreducible, hence by the above theorems the proof follows. □

Note that normalization is not necessary for convergence, but is helpful for numerical stability. We show that normalization preserves the eigenvector property.

Lemma 27. *Given a positive homogeneous function* $U : \boldsymbol{x} \mapsto \boldsymbol{y} \quad \boldsymbol{x}, \boldsymbol{y} \in {\mathbb{R}_+}^N$*, the function* $\frac{U}{||U||_1}$ *is positive and homogeneous with degree 0 and maps to the range* Δ^{N-1}*, keeping normalized eigenvectors invariant.*

Proof. $U(c\boldsymbol{v}) = c^\alpha U(\boldsymbol{v})$ hence $\hat{U}(c\boldsymbol{v}) = \frac{c^\alpha U(\boldsymbol{v})}{||c^\alpha U(\boldsymbol{v})||_1} = \frac{c^\alpha U(\boldsymbol{v})}{c^\alpha ||U(\boldsymbol{v})||_1} = \hat{U}(\boldsymbol{v})$. Now let $\boldsymbol{u}$ be a eigenvector of U, i.e. $U(\boldsymbol{u}) = \lambda \boldsymbol{u}$. Then $\hat{U}(\frac{\boldsymbol{u}}{||\boldsymbol{u}||}) = \hat{U}(\boldsymbol{u}) = \frac{\lambda \boldsymbol{u}}{\lambda ||\boldsymbol{u}||} = \frac{\boldsymbol{u}}{||\boldsymbol{u}||}$. □

Note however, it is not enough for the updates to have a unique fixed point, one also requires that the fixed point be stable, i.e. if perturbed from the fixed point value the updates will converge back to the fixed point. The stability issues are investigated in the following theorem, together with the question does any initialization followed by iterative re-mapping reach the fixed point.

We draw intuition from the Perron-Frobenius theorem which explores the same questions for positive matrices which are nothing but linear functions. Irreducibility of a matrix ensures that a change in any component of the vector propagates to all components of the vector when repeatedly multiplied by the matrix. One also requires for convergence that the weighted graph of the matrix obtained by interpreting it as a adjacency matrix be free of isolated cycles. In the following part of the article we will see that irreducibility followed by acyclicity (aperiodicity) is sufficient for convergence to a fixed point initialized by any positive vector. We also make the following note that if there exists a iteration number after initialization such that the iterate vector is strictly greater than the initialization it cannot have cycles. This inequality condition is called "primitivity" and is equivalent to aperiodicity. We show that if the function is primitive, iterations with any semi-positive vector will converge to a fixed point.

Definition 3. *A function U is defined to be primitive at $\boldsymbol{x}$ if $\forall_{\boldsymbol{y}\geq\boldsymbol{x}}\ \exists t \quad \text{s.t.} \quad U^t(\boldsymbol{y}) > U^t(\boldsymbol{x})$. If the function is primitive everywhere in its domain it is globally primitive* [2]

Theorem 14. *A positive increasing homogeneous function of degree ≤ 1, and a positive increasing super homogeneous function of degree < 1, that is irreducible and globally primitive has a unique fixed point to which iterations from any semi-positive initialization converges.*

Proof. Let $M(\boldsymbol{x},\boldsymbol{y}) = \text{Max}_i \frac{x_i}{y_i}$ and $m(\boldsymbol{x},\boldsymbol{y}) = \text{Min}_i \frac{x_i}{y_i}$ and let $r(\boldsymbol{x},\boldsymbol{y}) = \frac{m(\boldsymbol{x},\boldsymbol{y})}{M(\boldsymbol{x},\boldsymbol{y})}$. Note that for $\alpha > 0, \beta > 0\ r(\alpha\boldsymbol{x},\beta\boldsymbol{y}) = r(\boldsymbol{x},\boldsymbol{y})$. Moreover, since $\text{Max}_i \frac{x_i}{y_i} \text{Max}_j \frac{y_j}{z_j} \geq \text{Max}_k \frac{x_k}{z_k}$ and by similar argument $\text{Min}_i \frac{x_i}{y_i} \text{Min}_j \frac{y_j}{z_j} \leq \text{Min}_k \frac{x_k}{z_k}$ we have a inequality $r(\boldsymbol{x},\boldsymbol{y}) + r(\boldsymbol{y},\boldsymbol{z}) \geq r(\boldsymbol{x},\boldsymbol{z})$.

Let us use the shorthand $\boldsymbol{x}^t$ to denote $U^t\boldsymbol{x}$, and consider any vectors $\boldsymbol{y}$ and $\boldsymbol{x}$ such that $\boldsymbol{y} \leq c\boldsymbol{x}$ note that there is no loss of generality involved as such a c can always be found. Because of primitivity there exists a t such that the after t iterations the inequality is strict, i.e. $\boldsymbol{y}^t < c\boldsymbol{x}^t$. Hence $M(\boldsymbol{x}^t,\boldsymbol{y}^t) < M(\boldsymbol{x},\boldsymbol{y})$ and $m(\boldsymbol{x}^t,\boldsymbol{y}^t) < m(\boldsymbol{x},\boldsymbol{y})$. Thus we have $r(\boldsymbol{x}^t,\boldsymbol{y}^t) < r(\boldsymbol{x},\boldsymbol{y})$ for the specific value of t.

Consider the sequence of numbers $r(\boldsymbol{x}^{(n+1)t},\boldsymbol{x}^{nt})$, clearly it is a reducing sequence lower bounded by 1 and hence has a limit. Because of triangle inequality $\boldsymbol{x}^{n+1}t$ converges to a fixed point. From irreducibility and monotonicity we have uniqueness. □

The proposition above indicates that the iterations are stable. Irrespective of any perturbation to a corrupted semi-positive vector, a sequence of iterations would converge.

We explore the special case that the index p is taken to ∞. The corresponding norm is then equivalent to choosing the maximum of all the normalized ranks of the in vertices. This is both computationally favorable and resistant to Sybil like attacks. Breaking away from the Concavo-Convex ranking framework, the above strategy maybe generalized so that one takes a generalized mean of some fixed top k of the incoming degree divided ranks. Although we have shown that the properties of L_p ranks are nice for $p \geq 1$, when

[2] The important thing to note is that the second inequality is strict.

p is taken to infinity some the underlying assumptions break down such as the property of irreducibility. This can however be easily fixed but we omit the details out of space constraints.

6.3 Propagation of Trust

In this section we show how it is theoretically possible to incorporate spam and non-spam labels on vertices if they are available. We mention that even without the use of such labels the L_{pq} ranks offer significant spam-resistance over Pagerank and near to that of Trustrank, equal split Demotedrank Wu et al. (2006) and AntiTrustrank Krishnan and Raj (2006).

Consider we have a small hand-labelled set of trustworthy vertices $\mathcal{V}_+$ and spam pages $\mathcal{V}_-$, the remaining pages are denoted by $\mathcal{V}_0$. We take the position that pages that are linked directly by the set $\mathcal{V}_+$ or through intermediate vertices should be rewarded by a value of trust decreasing with distance from $\mathcal{V}_+$. Similarly vertices that link to $\mathcal{V}_-$ directly or indirectly are to be punished.

A point worth paying attention to is that trust and distrust are made to propagate in *opposite* directions. A page is rewarded based on what other pages think of it (i.e. through endorsement by nodes that it cannot control) on the other hand it is punished based on the links it has control over. It would be unfortunate for a page to be penalized because of an untrustworthy page that points to it as an act of malice. This is the approach followed in Wu et al. (2006), Krishnan and Raj (2006).

The reward and punishments are allocated based on the Concavo-Convex ranking function, with an exponential decay factor depending on the number of hops the vertex is away from the labeled sets. Let $\boldsymbol{s}_+$ be a vector that has 1s in place of the ranks of $\mathcal{V}_+$ and 0 otherwise, i.e. $s_+(i) = \mathbf{1}(i \in \mathcal{V}_+)$ and similarly $\boldsymbol{s}_-$ be a vector that has 1s in place of the ranks of $\mathcal{V}_-$ and 0 otherwise. Considering a decay parameter γ (for simplicity we take it to be the same for both directions) the reward of the set of vertices one hop distance away is $\gamma U(\boldsymbol{s}_+)$, the reward of those that are one hop distance from these points is $\gamma^2 U(U(\boldsymbol{s}_+))$,

generalizing to k hops which is $\gamma^k U^k(\boldsymbol{s}_+)$. Here we abuse the notation $U^k(\boldsymbol{s}_+)$ to mean k composition of the function U. The total reward is provided by the function

$$\boldsymbol{r}_+ = \sum_{\{0 \leq k\}} \gamma^k U^k(\boldsymbol{s}_+) \tag{6.3}$$

Lemma 28. *For* $\gamma < \lambda$ $Eig\left(U\boldsymbol{r}_+ - \gamma U(\boldsymbol{r}_+)\right) = \boldsymbol{s}_+$.

Proof. Apply the operator $[I - \gamma U]$ on both sides of equation 6.3 to obtain: $[I - \gamma U]\boldsymbol{r} = \sum_k \gamma^k [I - \gamma U] U^k(\boldsymbol{s}_+)$ or

$$\boldsymbol{r} - \gamma U(\boldsymbol{r}) = \sum_k [\gamma^k U^k(\boldsymbol{s}_+) - \gamma^{k+1} U^{k+1}(\boldsymbol{s}_+)] = \boldsymbol{s}_+ \tag{6.4}$$

The last line follows from the assumption that the k-th term in the tail of the summation converges to $\lambda^k \boldsymbol{x}$ where $\boldsymbol{x}$ is the nonlinear eigenvector of U □

Similarly considering $U^\dagger$ to be the function applied to the outlinks instead of the inlinks, one obtains $\boldsymbol{r}_- = \sum_{\{0 \leq k\}} \gamma^k {U^\dagger}^k(\boldsymbol{s}_-)$.

Lemma 29. *For* $\gamma < \lambda$ $Eig\left(U\boldsymbol{r}_- - \gamma U^\dagger(\boldsymbol{r}_-)\right) = \boldsymbol{s}_-$.

The difference in the computed $\boldsymbol{r}_+$ and $\boldsymbol{r}_-$ will indicate the level of trustworthiness.These can be combined in different ways. We choose $\frac{r_+}{r_+ + r_-} r$ to be the rank value with which the pages are evaluated. We need to provide a computational recipe for computing $\boldsymbol{r}_+$ and $\boldsymbol{r}_-$. For these we use the equations

$$\boldsymbol{r}_+^{t+1} = \gamma \hat{U}(\boldsymbol{r}_+^t) + \boldsymbol{s}_+ \text{and } \boldsymbol{r}_-^{t+1} = \gamma \hat{U}^\dagger(\boldsymbol{r}_-^t) + \boldsymbol{s}_- \tag{6.5}$$

The rank of a vertex is scaled by the trust and distrust ranks as $r = \frac{r_+}{r_+ + r_-} r$. For the purpose of the chapter, trust and distrust will always be used to obtain the ranks as a linear scaling as indicated. The ranks r_+ and r_- are the ranks the page receives from the vertices in $\mathcal{V}_+$ and $\mathcal{V}_-$, thus the form of scaling has an implicit assumption that the make

up of the remaining rank of a page received from other vertices also have the same spam versus non-spam ratio.

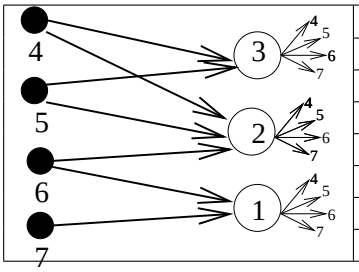
6.4 Experiments

We conducted experimental evaluation of the proposed ranking scheme on a publicly available, real-world and collaboratively labeled dataset[3]. It involves a 0.4 million vertex subset of the webgraph containing 13 million edges. This was the largest corpora collected for the web spam challenge-2007 web (2007) called "Large Dataset, track II".

Before we present results on the web-graph data set, we investigate the effects of the proposed algorithms on a few toy graphs where unlike the former we can identify, study and isolate the effects.

6.4.1 Results on Toy Graphs

These results are included solely to benefit our understanding of the effects that the proposed updates induce. More complete examination of spam-resistance of the proposed method is demonstrated on a portion of the real web-graph, right after these results on toy graphs.



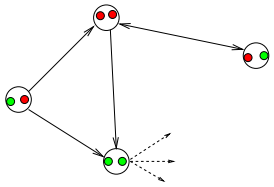
No.	L_1	L_2	L_∞
1	**0.183**	**0.210**	**0.211**
2	**0.183**	**0.165**	**0.160**
3	0.129	0.138	0.108
4	0.126	0.122	0.130
5	0.126	0.122	0.130
6	0.126	0.122	0.130
7	0.126	0.122	0.130

Figure 6.2: Example Graph - I, vertices {1,2,3 } are connected to {4,5,6,7} by edges, not shown for clarity. Demonstrates property 1 of f^i for L_p and Pagerank. Details in text.

Consider the graph depicted in figure 6.2, we have not drawn the edges from the vertices {1,2,3 } to the vertices {4,5,6,7} to avoid clutter. Because the vertices {4,5,6,7} have identical inlinks their ranks are identical. Vertex 2 receives 3 links each worth 1/2,

[3]We thank the organizers of the webspam challenge for making such a difficult to obtain data available

whereas vertex 1 receives links worth 1 and 1/2 respectively. Since the total flow received by 1 and 2 are the same, they are ranked equally by the Pagerank algorithm. In order to reduce spam susceptibility, we desired that pages that receive multiple low quality links be ranked lower than those that receive links from a few high quality pages, even if their total Pagerank flow is the same. This property is exhibited by the L_2 and L_∞ rankings as shown. In figure 6.3 we investigate a link-farm spam scenario. In order to spam node 6, one has



No	L_1	L_2	L_∞
1	2.082e-1	2.074e-01	3.850e-1
2	1.645e-1	1.470e-01	2.371e-1
3	6.092e-2	7.171e-02	8.333e-2
4	6.092e-2	7.171e-02	8.333e-2
5	6.092e-2	7.171e-02	8.333e-2
6	**2.132e-1 (1)**	**1.825e-01 (3)**	**6.030e-2 (6)**
7	7.708e-2	8.260e-02	2.248e-2
8	7.708e-2	8.260e-02	2.248e-2
9	7.708e-2	8.260e-02	2.248e-2

Figure 6.3: Example Graph - II. The dark nodes are taken to be legitimate vertices, whereas node 6 is being spammed by nodes {7,8,9} that are otherwise disconnected from the graph. Vertex 1 connects out to all dark nodes, as does vertex 6 to all white nodes. Also shown in this figure are the (spammed) Pagerank and L_P Rank scores, together with the ranks of the node 6.

created vertices {7,8,9} to point towards it. Even without links from the tightly connected larger legitimate web like network of black nodes, the Pagerank of 6 is the highest, thereby showcasing its vulnerability. The significance of this example is that the subgraph structures of the form $\{6, 7, 8\}$ are commonly used to spam the Pagerank. The L_p Rank algorithms can be seen to be more resistant to this. It is not unspammable but would take orders of magnitude more pages to do so. Spam like pages are demoted depending on the value of p.

The best result in terms of quality is offered by L_∞ Rank, in which case the spammed page gets ranked below all the legitimate pages.

6.4.2 Results on Real Web-Graph

In this section we describe our experiments on a 0.4 million vertex subset of the webgraph containing 13 million edges that was human labeled and made publicly available for the Spam-challenge-2007. This data set was the largest corpora collected for the web spam challenge-2007 web (2007) called "Large Dataset, track II". About 80% of the web pages of this corpus are non-spam whereas the remaining are labelled as spam. All labels were generated as a collaborative effort involving several human evaluators.

We evaluate L_{pq} rank on a number of metrics and compare it with other benchmarks, namely, normalized in-degree, Pagerank, Trustrank, equally split Demotedrank (equivalently to AntiTrustrank) and also L_p. Note that in-degree is defenseless against linkspam attacks but is included as a benchmark, because it has been observed to correlate well Najork et al. (2007) with quality of a page (perhaps because spammers do not target it any more). Human perceived quality of the rankings induced on this data set cannot be evaluated because of its anonymized nature. The corpus consists of the adjacency matrix of the graph as well as a tf-idf representation of its contents. Both the identity of the pages as well as that of the features are anonymized in order to prevent web-spam challenge participants from using extraneous information from the web for the task. Neither the identity of the page nor the contents of the page can be retrieved. A side effect of which is that user studies are not possible. Though (anonymized) tf-idf features were available, we focussed on spam resistance that can be extracted from the link structure alone. Recall that we are not competing with content feature based spam classifiers. While they are easy to train, spammers are also free to change the content at will to counter it. Topological properties of the webgraph, on the other hand is relatively harder to manipulate.

Due to lack of an agreed upon gold standard ranks of the vertices, evaluation of a

ranking function is contentious and certain assumptions have to be made about what constitutes a good rank. There will always be disagreement on the adequacy and completeness of any such characterization, but more the number of criteria according to which a vertex is ranked higher above the rest, more confidence one would have regarding its goodness. We mention what our assumptions are and what we consider to be a "good" rank and how we measure the multi-criteria quality of a ranking. We consider both the ordinal rank as well as the rank score values for evaluation, because both are important. Since the total probability mass or rank-score assigned by our ranking algorithm equals 1, a quality measure that we look at is how much of the total "probability mass" does L_p and L_{pq} rank assign to the spam pages. This mass is compared with the probability mass assigned by Pagerank (or equivalently L_1 rank), normalized in-degree and AntiTrustrank. The lower this mass for a scheme, the better it is. This measure is more complete than counting the number of spam pages occurring in a top-K ranked list for some fixed low value of k . A low total probability mass indicates that on average non-spam pages are ranked higher. There is one situation where this measure can fail, that is if the ranking scheme allocates almost all of its mass to some good site and near negligible to all the rest. To ensure that this is not happening in practice, we include another ordinal measure: curves of the number of spam pages encountered as one traverses down the rank order, starting from 1 to the total number of pages N. Ideally all spam pages should come last. The closer this curve is to the X axis the better is the ranking function.

It is not enough for a ranking scheme to just assign low mass to spam pages. The ranks induced on the non-spam pages has to be of high quality, and this is what differentiates a ranking scheme from a classifier. Since we do not have a standard rank ordering, we computed rank distance measures between our parametric family of ranks and other baseline algorithms, such as Pagerank, (Anti)Trustrank and in-degree, on the non-spam pages. The rank distance measures that is used is Spearman's foot rule statistics Diaconis and Graham (1977). If $R_1()$ and $R_2()$ are two rankings induced on a set $\mathcal{X}$, i.e. R_1 and

R_2 take integer values in $1, |\mathcal{X}|$, then the Spearman's foot rule distance between R_1 and R_2 is defined as $spearman(\boldsymbol{R_1}, \boldsymbol{R_2}) = \sum_{\mathcal{X}} |R_1(x) - R_2(x)|$. We use the normalized version $\frac{\sum_{\mathcal{X}} |R_1(x) - R_2(x)|}{|\mathcal{X}|^2}$. Apart from the quality measures described above we also looked at speeds of convergence.

The organizers of the challenge had identified a 10% fraction of the vertices to be used for training and cross-validation, the remaining for testing. Our algorithm is not a learning algorithm and does not have a training phase, and our results are for the *unsupervised scenario*. However we do compare it with Trustrank Gyongyi et al. (2004), and AntiTrustrank equivalents Wu et al. (2006), Krishnan and Raj (2006) which are algorithms that take into account spam and non-spam labels on a training set of vertices. We thus report a second group of experiments where the training vertices were used as a seed set for propagating trust and distrust values to affect the ranking much like Trustrank. For this labeled case, we used the small label set identified by the organizers to seed the propagation of trust and distrust as in equations (6.5). Here the baseline is stronger and is (Anti)TrustRank algorithm. (Anti)Trustrank is that analogue of Pagerank that uses the flow of trust/distrust.

Before discussing the results obtained by the propagation of trust, we would like to draw the reader's attention towards an important point regarding the vulnerability of Trustrank that has also been alluded to in the introductory section. Trustrank evaluates the trust and untrustworthiness of a page from its distance from labelled "good" and "spam" pages. A page to which a "good" page points, accrues trust, whereas a page to which a spam page points accrues distrust. The latter is problematic because it allows a page to maliciously point to any page and demote its rank. This can easily be fixed, if distrust is propagated in reverse that is, a page accrues distrust if the evaluated page points to a spam page. The trust model with this reversed direction of flow of distrust is called the Opp(osite) Trustrank model in the experiments. On the data set it fairs somewhat worse than the original Trustrank flow of distrust, but that is because on the subgraph of the web captured by the data set, the spammers have not exploited this loophole. The performance

of Trustrank should hence be taken with due consideration, because it is unusable on the real internet.

For the propagation of trust model, we conducted separate experiments for the two directions of flow of distrust. The TrustRank formulation takes the direction to be same as that of the graph. We observed that this direction has a better discriminative property to separate spam and non-spam. However this direction of flow can be exploited with malicious intent and should not be used in practice.

Here we investigate the behavior of L_p ranks and establish it as a strong baseline bettered subsequently by the $L_{p,q}$ ranks both in quality and speed of convergence. The benefit of L_p over Pagerank is moderate and is discussed only as an example, it is $L_{p,q}$ ranks that perform strikingly better, both in terms of spam resistance and speed of convergence. Hence we propose their use.

The probability mass assigned to the spam pages when running L_p rank algorithm on the webspam-challenge graph is shown in figure 6.4 along with the horizontal curve indicating the performance of ranking by *in-degree* that achieves a spam mass of 25.28%. Note that indegree ranking on this data set is worse than Pagerank and it is very susceptible to spam attacks. An important observation is the low spam discriminative property of Pagerank. The spam mass of 0.2067 for Pagerank is of the same order of magnitude as the amount of spam in the entire data set (0.20). With increasing values of p the spam mass reduces by 40%. Plotted together with the L_p rank masses are two other curves, one for L_p trust rank which is the L_p generalization of Trustrank as explained in section 6.3, the other for the same except that distrust is made to flow in the opposite direction. L_p Tr.Rank corresponds to propagation in the same direction as the edges whereas L_p Opp Tr.Rank has opposite direction of flow of distrust. Though same direction propagation of trust performs better for most values, under this scheme a page is open to malicious attacks from a untrustworthy page as mentioned before, and is hence un-usable in practice.

Pairwise normalized Spearman footrule distances between the L_p rankings are shown

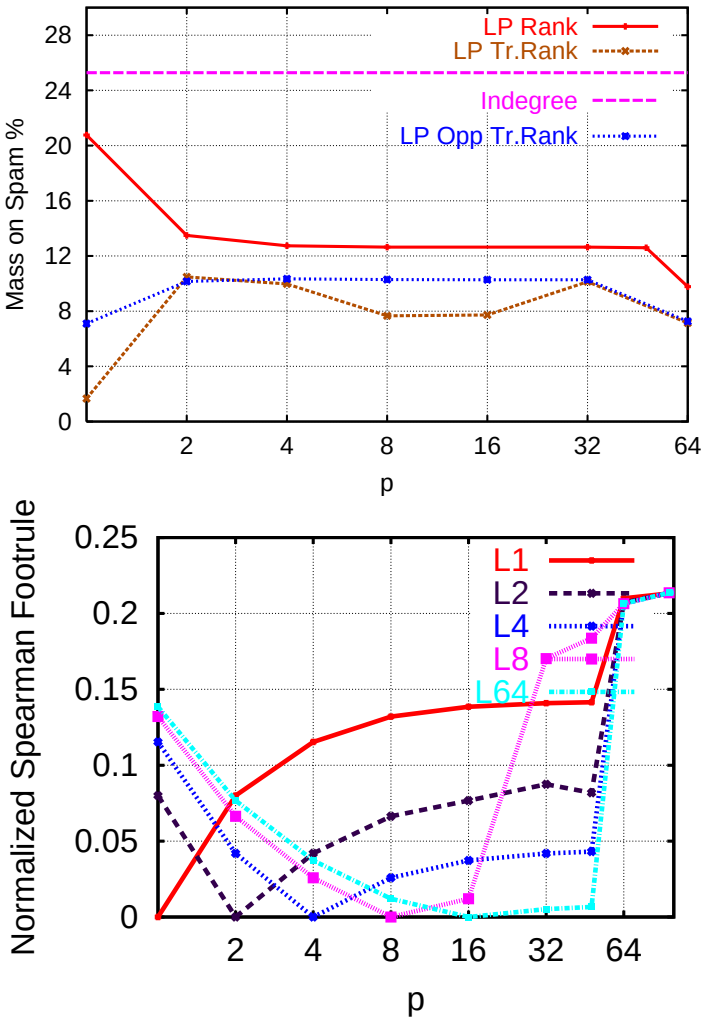


Figure 6.4: **Top:** Probability mass assigned by L_p ranks and in-degree rank on the spam pages. **Bottom:** Spearman footrule distance between different rankings on the spam pages.

in figure 6.4 (to the right) together with comparison with the order induced by Pagerank on non-spam vertices. One can observe that ranks that are close in p are also close in Spearman's foot rule distance, however one can see that L_p ranks are close to the Pagerank (L_1) rank order. This confirms that L_{pq} ranking largely agrees on the non-spam vertices, the agreement is higher with Trustrank than with Pagerank.

Rates of convergence at different values of p are shown in figure 6.5, the rate settles

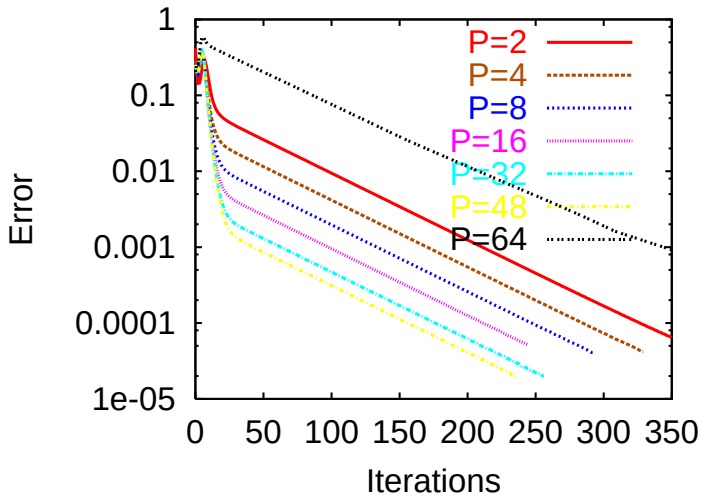


Figure 6.5: Rates of convergence of absolute error between consecutive iterates of L_p algorithm with uniform initialization. Compare this baseline with the improved convergence rates for L_{pq} **Figure 6.8**

into a constant exponential decay after an initial unstable domain. For L_p rank the basic fixed point iterations are too slow (unlike L_{pq} to be described next) and a constant linear damping was added for speed up. Even with the linear damping, convergence required several hundreds of iterations (unlike Pagerank which converged under 50 iterations). This should be compared with the superlative convergence rates obtained for the L_{pq} updates shown in figure 6.8.

We propose the use of L_{pq} algorithm. Recall that $q > p$ leads to convergent and unique ranks, experiments were conducted in this setting. It is observed that the value of q has a very significant role to play on rank quality and the convergence rate. The number of iterations required, drops monotonically from several hundreds of iterations to few tens of iterations as shown in figure 6.6 making the scheme a practical proposition. The effect of q on spam reduction is such that there is a best value of q at which the spam reduction is the highest, it was empirically observed to lie close to p as shown in figure 6.6, bottom. The value of q thus plays a crucial role to obtain a fast algorithm with good unsupervised spam fighting capabilities.

We investigate in detail the behavior of $L_{p,q}$ ranks at $\frac{q}{p} = 1.2$, that was found to have a good trade off between speed and spam resistance, and varying values of p. The particular value of the fraction was chosen from the preliminary experiments shown in 6.6. The probability mass assigned by $L_{p,q}$ ranks and Trust and Opp.Trust $L_{p,q}$ ranks on the spam pages are shown in figure 6.7. Note that how the unlabelled $L_{p,q}$ rank at $p = 4$ matches the performance of $L_{p,q}$ Opp.Trustranks at $p = 1$. Furthermore Opp.Trust model performs equally well as the conventional Trust model. For L_p ranks Opp.Trust models performed worse.

The convergence behavior is shown in figure 6.8, all the values of p show very rapid exponential rate of convergence and around 40 iterations is sufficient, Pagerank too takes about 50 iterations to converge. Pairwise normalized Spearman footrule distances between the $L_{p,q}$ rankings are shown in figure 6.7 together with the spearman footrule distance of the ranks induced by in degrees, recall $p = 1$ corresponds to Trustrank. From the graph one can observe that the $L_{p,q}$ ranks are close to those induced by Trustrank on non-spam pages and very close to each other. One can also observe that for values of $p = 4$ and higher, the rank order is almost the same. The ranks for $p = 2$ is closer to Trustrank than those for $p = 4$ and higher. On the other hand the ranks induced by the in-degrees are are distant in normalized Spearman footrule distance sense from the $L_{p,q}$ rankings. In fact we were able to verify that most of the pages ranked high by the in-degree were spam pages, see figure 6.10.

The best results were obtained for the family where the trustworthiness and untrustworthiness were propagated linearly whereas the basic rank used $L_{p,q}$ nonlinearity. This setup is named the Lin-L_{pq} variant. On retrospect its performance is easy to explain. The hand labels of (spam and non-spam) are of high quality and are not targeted by spam. Thus there is no reason to use the nonlinear generalization to counter "label" spam. The best spam resistance performances are shown below. The figure 6.9 shows the amount of probability mass assigned by the Lin-$L_{p,q}$ variant. The horizontal lines indicate the probability

mass assigned by Trustrank and AntiTrustrank. One can see that Lin-L_{pq} convincingly outperforms the best performance seen so far. Note that the Y-axis is log-scaled for better resolution and the gap in the performances is higher than it looks. The figure 6.10 shows the precision recall curves for Pagerank and Lin-$L_{p,q}$ variant with reversed flow of distrust. The plots for same direction of flow of trust have the same nature as these and had to be omitted to save space. On the Y axis it plots the number of spam pages encountered with decreasing Lin-$L_{p,q}$ rank. Nearer the curve is to the X axis the better the algorithm and a diagonal line indicates that spam and non-spam occur with equal frequency. From the plot corresponding to Pagerank and its deviation from the diagonal it is possible to note that though Pagerank allocates about the same total probability mass to spam as the total percentage of spam vertices, the spam pages occur towards lower ranked pages. However Pagerank performance is overwhelmingly outperformed by the Lin-$L_{p,q}$ variants. The same plot is shown drawn to log-scale to the right for better resolution because the L_{pq} family curves are almost indistinguishable from the X axis. From the log-scale plot one can observe that for $p = 4$ and higher the curves almost overlap, $p = 2$ has less spam initially but crosses the other set of curves. Thus a strategy that chooses between these two cases depending on the rank may be effective. The cumulated spam curves are compared with the cumulated curve induced by rankings based on in-degree, see figure 6.10. One can observe that the ranks based on in-degree have the worst characteristic among all the rankings considered, faring significantly worse than Pagerank, which the Lin-$L_{p,q}$ family beats convincingly.

Figure 6.9 establishes the fact that it is better to use the LinL_{pq} ranks over Trustrank when labels are available. The main difference between the LinL_{pq} rank and Trustrank is that the former uses non-linear updates for the propagation of the rank score whereas the latter uses a linear propagation. The flow of trust and distrust are however linear for both. Now an important question arises regarding the number of labeled examples required by the two methods in order to give equivalent spam resistance performance. This is explored next.

We include a plot that compares the spam detection properties of the proposed family and TrustRank at different percentage of labels available. For the comparison two simple threshold based spam classifier were learnt using the L_{pq} Trust rank and the TrustRank values as their corresponding single feature. The classification error rates are shown for different labelled set sizes and the optimal threshold, see figure 6.11. From this one can observe that LinL_{pq} can provide superlative spam resistance at a fraction of the number of labels required by Trustrank.

6.5 Conclusion

We propose a large family of link-analytic ranking functions based considerations of spam resistance, convergence and initialization independence. It is remarkable that convergence guarantees can be carried over to the nonlinear ranking functions. Properties of a parametric subfamily that includes Pagerank and Norm() as a special case was studied in detail, both theoretically and experimentally. Appropriate choice of the ratio p/q gives excellent spam resistance on the internet graph when used with and without labels.

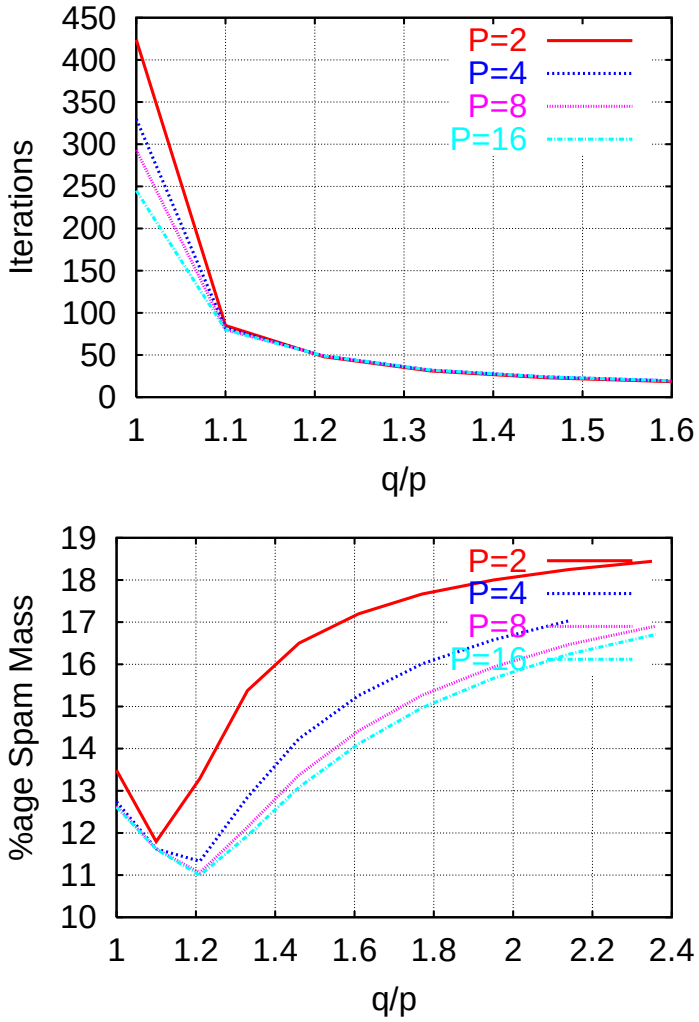


Figure 6.6: Top: Iterations required for convergence (absolute error between consecutive iterates less than 1e-6) of $L_{p,q}$ algorithm with uniform initialization with p held constant and increasing q Bottom: Probability mass assigned to spam for the same

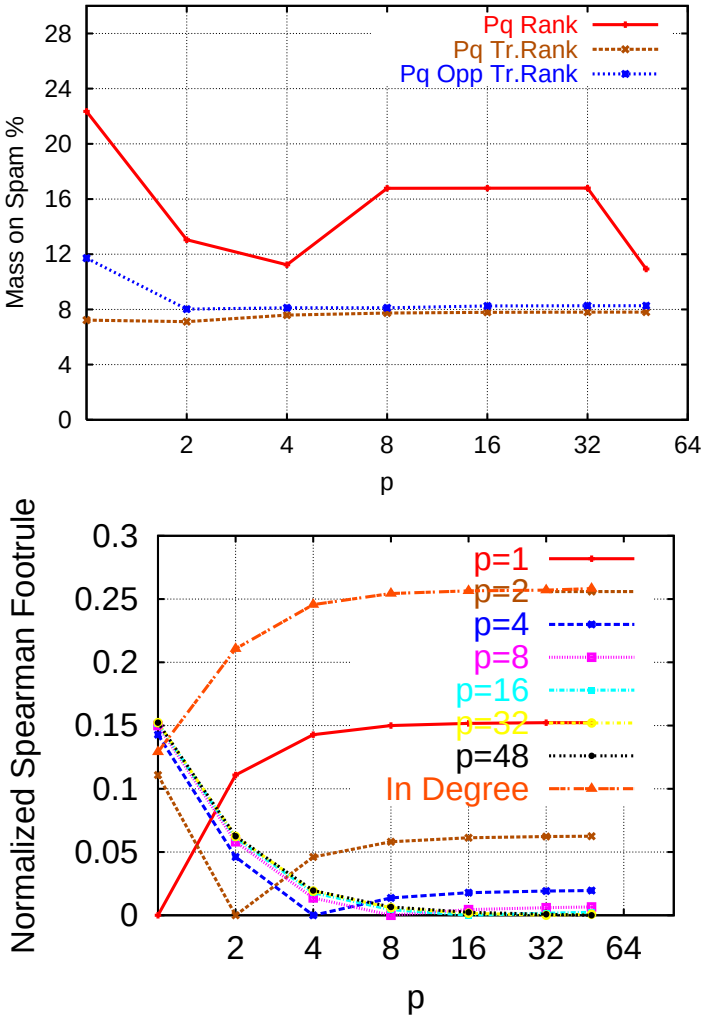


Figure 6.7: Top: Probability mass assigned by $L_{p,q}$ ranks on the spam pages for $\frac{q}{p} = 1.2$. Bottom: Corresponding Spearman footrule distance between different $L_{p,q}$ and In-degree rankings.

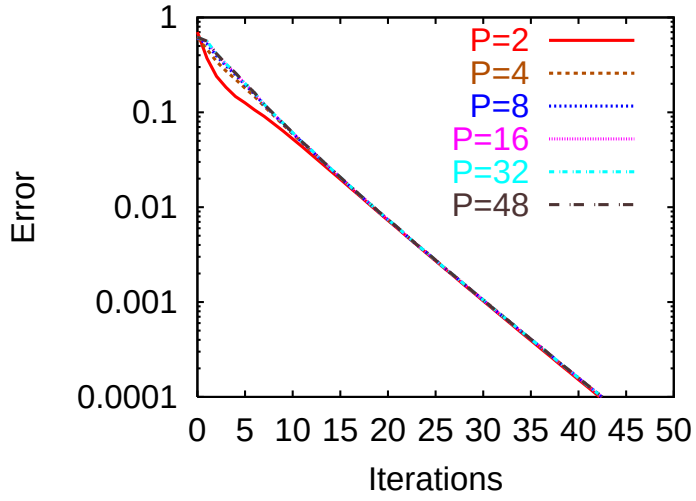


Figure 6.8: Rate of convergence of absolute error between consecutive iterates of L_{pq} Rank algorithm initialized with uniform ranks for $\frac{q}{p} = 1.2$

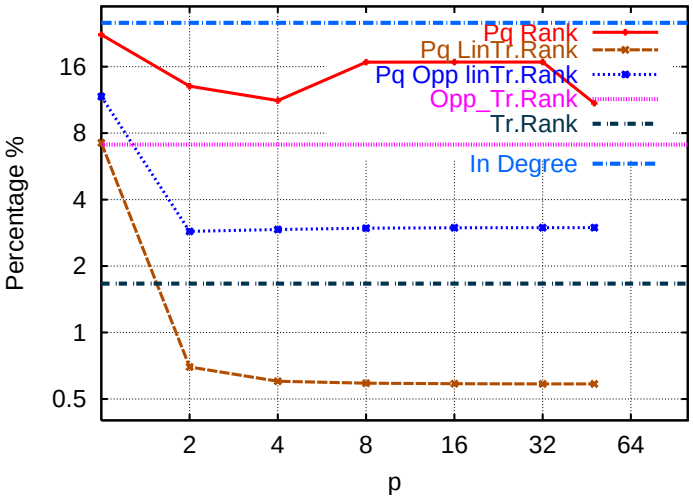


Figure 6.9: Probability mass assigned by Lin-Pq and in-degree rank (25.28%) on the spam pages.

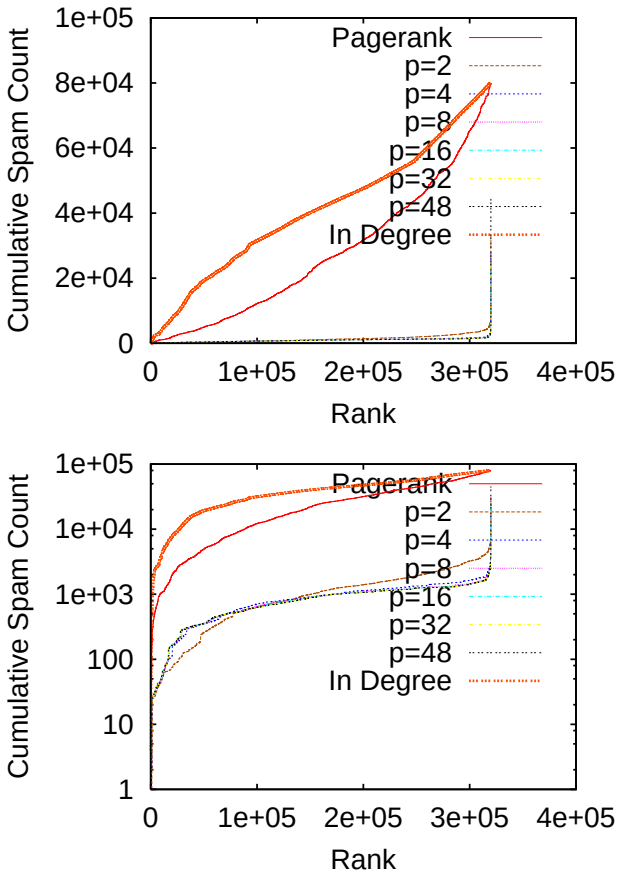


Figure 6.10: Top: Number of spam encountered in sorted rank order for Lin-*pq* ranks. Y axis measures number of spam pages and X axis decreasing rank. Bottom: The same log-scaled

Algorithm	0.625%	1.25%	2.5%	5%	10%
L_{pq} TrustRank	7.548	5.646	4.067	3.167	2.712
TrustRank	65.103	64.141	63.466	15.576	14.57

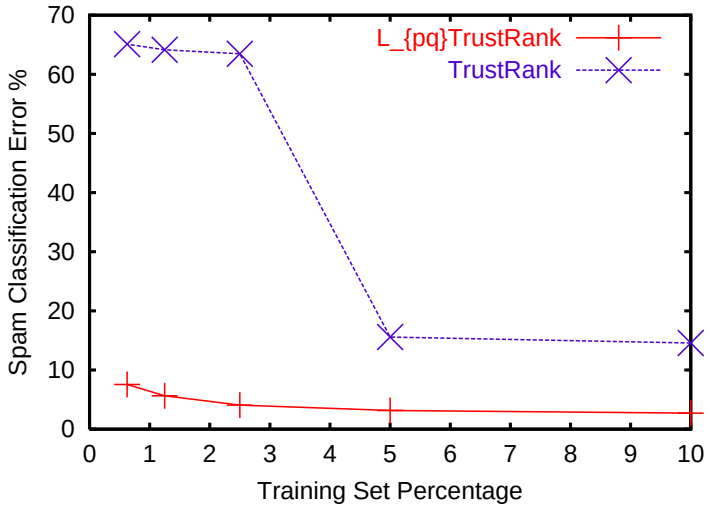


Figure 6.11: Spam classifier error rates for a single feature classifier at different number of training vertices.

Chapter 7

Conclusion

In this dissertation we addressed different aspects of learning to rank in both supervised and unsupervised settings. Monotonic transformations form a natural framework to pose ranking problems in because they preserve order. Modeling, manipulating and exploiting monotonic transformations played a key role in all of the aspects of the problems covered in this dissertation. The first part of the dissertation was on building tools that allow efficient optimization of a loss function over this class of functions, without imposing any finite dimensional parameterization on them. This was greatly facilitated by the intimate connection between monotonicity, convexity and properties of minimizers of Bregman divergences constrained to lie on the monotone cone.

The ability to efficiently optimize a loss function over the class of monotonic transformations was extended to Bregman divergence based loss functionals whose gradient matches the monotonic transform. This guaranteed that the cost functional remained convex jointly in the space of functions and parameters ensuring global minimum. It also directly enabled learning the parameters of a canonical generalized linear model with an unknown link function, leading to substantial generality at the cost of worsening the time complexity of an iteration by only a logarithmic factor. The framework presented does not require one to pick one member from the infinite family of canaonical generalized models, since the

approach simultaneously optimizes over the choice of the family and the parameters of the family.

A large family of link-analytic, fixed point based ranking functions were proposed based on considerations of spam resistance, convergence and initialization independence. Here, again monotonicity and convexity played a key role. It is natural to desire that item A outrank item B if the recommendations/inlinks of A majorizes the recommendations/inlinks of B. This together with the notion that the order of recommendations/inlinks are irrelevant to the rank-score, determines that the ranking function is Schur convex. We used concavity to model the phenomenon of diminishing returns as more and more recommendations are received. Pagerank was shown to be reltaively susceptible to spam as it lacks strict Schur convexity and concavity. We showed that if we chose the ranking function to have Schur convexity, concavity and in addition be homogeneous of a certain degree, not only is the ranks determined by the fixed point unique but also that they can be reached using fixed point updates using arbitrary initialization.

It was also shown that pagerank, a successful unsupervised ranking method, can be looked upon as optimizing the consensus among several local recommendations over a set of items. This optimization view point then naturally enabled the formulation to be extended to the setting where there is fluctuation and uncertainty in the local recommendations. Since in the pagerank setting a recommendation map directly to edges in a graph, the formulation easily captures multiple and changing labels on the edges of the graph.

Appendix A

Proofs from Chapter 3

Notation: Follows notation of Chapter 3.

To prove Theorem 1 we will need the following lemma

Lemma 30. *Rockafellar (1996) Let the function $\phi(\cdot)$ be continuously differentiable and convex. If $s||\nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y})|| \leq ||\boldsymbol{x} - \boldsymbol{y}||$ then*

$$\phi(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \geq \alpha\phi(\boldsymbol{x}) + (1-\alpha)\phi(\boldsymbol{y}) - \frac{\alpha(1-\alpha)}{2s}||\boldsymbol{x} - \boldsymbol{y}||^2$$

Theorem 1

Proof. Let us introduce the abbreviations:

$$\begin{aligned}
\boldsymbol{x}(\alpha) &= \alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2 \\
\boldsymbol{y}(\alpha) &= \alpha\boldsymbol{y}_1 + (1-\alpha)\boldsymbol{y}_2, \\
\phi_i &= \phi(\boldsymbol{x}_i), \psi_i = \psi(\boldsymbol{x}_i), \\
\Phi(\alpha) &= \alpha\phi_1 + (1-\alpha)\phi_2 \\
\Psi(\alpha) &= \alpha\psi_1 + (1-\alpha)\psi_2.
\end{aligned}$$

To show joint convexity of the Fenchel Young gap, we have to show

$$\phi(\boldsymbol{x}(\alpha)) + \psi(\boldsymbol{y}(\alpha)) - \langle \boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha) \rangle \leq \Phi(\alpha) + \Psi(\alpha) - \alpha \langle \boldsymbol{x}_1, \boldsymbol{y}_1 \rangle - (1-\alpha) \langle \boldsymbol{x}_2, \boldsymbol{y}_2 \rangle$$
$$\forall\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in \operatorname{dom} \phi,\ \boldsymbol{y}_1, \boldsymbol{y}_2 \in \operatorname{dom} \psi.$$

or equivalently, show:

$$\phi(\boldsymbol{x}(\alpha)) + \psi(\boldsymbol{y}(\alpha)) \leq \Phi(\alpha) + \Psi(\alpha) + \overbrace{\alpha(1-\alpha) \langle \boldsymbol{x}_1 - \boldsymbol{x}_2, \boldsymbol{y}_1 - \boldsymbol{y}_2 \rangle}^{B}$$
$$\forall\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in \operatorname{dom} \phi,\ \boldsymbol{y}_1, \boldsymbol{y}_2 \in \operatorname{dom} \psi. \quad \text{(A.1)}$$

Assume with no loss in generality that $\phi(\cdot)$ and $\psi(\cdot)$ are strongly convex with modulus of strong convexity $(1+s1), (1-s2)$ with $s1 \geq -1$, $s2 < 1$, respectively.

From $(1+s1)$strong convexity of ϕ we have:

$$\langle \nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \geq (1+s1)||\boldsymbol{x} - \boldsymbol{y}||^2,$$
$$\text{or, } ||\nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y})|| \geq (1+s1)||\boldsymbol{x} - \boldsymbol{y}|| \quad \text{(A.2)}$$

the second inequality follows from Cauchy Schwarz inequality. Similarly from $(1-s2)$ strong convexity of $\psi = \phi^*$ we have

$$\langle \nabla\psi(\boldsymbol{u}) - \nabla\psi(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v} \rangle \geq (1-s2)||\boldsymbol{u} - \boldsymbol{v}||^2,$$
$$\text{or, } \left\langle (\nabla\phi)^{-1}(\boldsymbol{u}) - (\nabla\phi)^{-1}(\boldsymbol{v}), \boldsymbol{u} - \boldsymbol{v} \right\rangle \geq (1-s2)||\boldsymbol{u} - \boldsymbol{v}||^2$$
$$\text{or, } \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y}) \rangle \geq (1-s2)||\nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y})||^2 \quad \text{(A.3)}$$
$$(1-s2)||\nabla\phi(\boldsymbol{x}) - \nabla\phi(\boldsymbol{y})|| \leq ||\boldsymbol{x} - \boldsymbol{y}|| \quad \text{(A.4)}$$

In (A.3) we have used $\boldsymbol{u} = \nabla\phi(\boldsymbol{x}), \boldsymbol{v} = \nabla\phi(\boldsymbol{y})$. From (A.4) and (A.2) we obtain

$$(1+s1)(1-s2) \leq 1. \quad \text{(A.5)}$$

Now, simplifying expression (A.1) using our strong convexity assumptions and positivity of $\alpha(1-\alpha)$, we reduce (A.1) to

$$(1+s1)||\boldsymbol{x}_1-\boldsymbol{x}_2||^2+(1-s2)||\boldsymbol{y}_1-\boldsymbol{y}_2||^2-2B\leq 0$$
$$\text{Or, } ||(\boldsymbol{x}_1-\boldsymbol{x}_2)-(\boldsymbol{y}_1-\boldsymbol{y}_2)||^2+s1||\boldsymbol{x}_1-\boldsymbol{x}_2||^2-s2||(\boldsymbol{y}_1-\boldsymbol{y}_2)||^2\leq 0.$$

Let $\boldsymbol{p}=\boldsymbol{x}_1-\boldsymbol{x}_2$ and $\boldsymbol{q}=\boldsymbol{y}_1-\boldsymbol{y}_2$. By choosing $(1+s)\boldsymbol{p}=\boldsymbol{q}$ we obtain $s1>s2+s1s2$, or equivalently $(1-s2)(1+s1)\geq 1$. From (A.5) we have $(1+s1)(1-s2)=1$. From (A.4) and Lemma 30 we obtain

$$\phi(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y})\geq\alpha\phi(\boldsymbol{x})+(1-\alpha)\phi(\boldsymbol{y})-\frac{1}{2(1-s2)}\alpha(1-\alpha)||\boldsymbol{x}-\boldsymbol{y}||^2$$

but by assumption (see (2.1))

$$\phi(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y})\leq\alpha\phi(\boldsymbol{x})+(1-\alpha)\phi(\boldsymbol{y})-\frac{1+s1}{2}\alpha(1-\alpha)||\boldsymbol{x}-\boldsymbol{y}||^2.$$

As we have already established $(1+s1)(1-s2)=1$, we have for $k=\frac{1+s1}{2}$

$$k\alpha(1-\alpha)||\boldsymbol{x}-\boldsymbol{y}||^2=\alpha\phi(\boldsymbol{x})+(1-\alpha)\phi(\boldsymbol{y})-\phi(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}).\tag{A.6}$$

Taking derivative w.r.t α on both sides of (A.6) and setting $\boldsymbol{y},\alpha=0$ it follows that $\phi(x)=k||\boldsymbol{x}||^2$ for some $k>0$ (ignoring affine terns.) The case $s2=1$ follows using continuity. □

A.1 Optimality of Means

Theorem 15. *(Banerjee et al., 2005) Let $\boldsymbol{\pi}$ be a distribution over $\boldsymbol{x}\in\operatorname{dom}\phi$ and $\boldsymbol{\mu}=\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\boldsymbol{\pi}}[\boldsymbol{x}]$ then the expected divergence about $\boldsymbol{s}$ is*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\boldsymbol{\pi}}\left[D_\phi(\boldsymbol{x}\Big|\Big|\boldsymbol{s})\right]=\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\boldsymbol{\pi}}\left[D_\phi(\boldsymbol{x}\Big|\Big|\boldsymbol{\mu})\right]+D_\phi(\boldsymbol{\mu}\Big|\Big|\boldsymbol{s}).\tag{A.7}$$

From non-negativity of Bregman divergence it follows that:

Corollary 7. *(Banerjee et al., 2005)* $\underset{\boldsymbol{x}\sim\boldsymbol{\pi}}{\mathbb{E}}[\boldsymbol{x}] = \underset{\boldsymbol{y}\in\operatorname{dom}\phi}{\operatorname{Argmin}} \underset{\boldsymbol{x}\sim\boldsymbol{\pi}}{\mathbb{E}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{y}\big)\right].$

Combining identity (2.4) and Corollary (7) we obtain

Corollary 8. *(Banerjee et al., 2005) Generalized mean* $\mu^\phi(\boldsymbol{x}) = (\nabla)^{-1}\phi(\underset{\boldsymbol{x}\sim\boldsymbol{\pi}}{\mathbb{E}}[\nabla\phi(\boldsymbol{x})])$
$= \underset{\boldsymbol{y}\in\operatorname{dom}\phi}{\operatorname{Argmin}} \underset{\boldsymbol{x}\sim\boldsymbol{\pi}}{\mathbb{E}}\left[D_\phi\big(\boldsymbol{y}\big\|\boldsymbol{x}\big)\right].$

Corollary 9. *If random variable* $\boldsymbol{x}$ *takes values in* $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ *with* $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ *then*

$$\underset{\boldsymbol{\mu}\in\mathcal{X}}{\operatorname{Argmin}} \underset{\boldsymbol{x}|\mathcal{X}}{\mathbb{E}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{\mu}\big)\right] \geq \underset{\boldsymbol{\mu}_1\in\mathcal{X}_1}{\operatorname{Argmin}} \underset{\boldsymbol{x}|\mathcal{X}_1}{\mathbb{E}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{\mu}_1\big)\right] + \underset{\boldsymbol{\mu}_2\in\mathcal{X}_2}{\operatorname{Argmin}} \underset{\boldsymbol{x}|\mathcal{X}_2}{\mathbb{E}}\left[D_\phi\big(\boldsymbol{x}\big\|\boldsymbol{\mu}_2\big)\right].$$

Appendix B

Proofs from Chapter 4

Notation: Follows notation of Chapter 4.

B.1 Large Deviation Bound for Exponential Family Densities with Uniformly Concave Entropy

Let the random variable y taking values in $\mathcal{Y} \subset \mathbb{R}^n$ have the exponential family density

$$P(y) = e^{\langle \boldsymbol{y}, \boldsymbol{\theta} \rangle - \phi^*(\boldsymbol{\theta})}.$$

The function $\phi^*(\cdot) : \Theta \mapsto \mathbb{R} = \int_{\mathcal{Y}} e^{\langle \boldsymbol{y}, \boldsymbol{\theta} \rangle}$ is the log partition function and its Legendre conjugate

$$\phi(\boldsymbol{\mu}) = \sup_{\boldsymbol{\theta} \in \Theta} \langle \mu, \boldsymbol{\theta} \rangle - \phi^*(\boldsymbol{\theta})$$

is its negative entropy. It is assumed that $\phi(\cdot)$ is uniformly convex, i.e.

Theorem 16. *If random variable y has exponential family density $e^{\langle \boldsymbol{y}, \boldsymbol{\theta} \rangle - \phi^*(\boldsymbol{\theta})}$ with negative entropy $\phi(\boldsymbol{\mu})$ uniformly convex with respect to norm $||\cdot||$ with modulus $\delta(\cdot)$ then for*

any bounded convex set $\mathcal{B}$

$$P(\boldsymbol{y} \notin \mathcal{B}) \leq e^{-\sup_{\boldsymbol{y}\in\mathcal{B}} \delta(||\boldsymbol{y}-\mathbb{E}[Y]||)}.$$

Proof. Consider any bounded, convex set $\mathcal{B}$ with the support function

$$\sigma(s) = \sup_{\boldsymbol{y}\in\mathcal{B}} \langle s, \boldsymbol{y}\rangle .$$

Now the indicator function $\mathbf{1}_{\mathcal{B}}(\boldsymbol{y}) = \begin{cases} 1 & \text{if } \boldsymbol{y} \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$ of the set $\mathcal{B}$ can be bounded as

$$1 - \mathbf{1}_{\mathcal{B}}(\boldsymbol{y}) \leq e^{\langle s,\boldsymbol{y}\rangle - \sigma(s)}.$$

Therefore

$$\begin{aligned} P(\boldsymbol{y} \notin \mathcal{B}) &\leq \mathbb{E}\left[e^{\langle s,\boldsymbol{y}\rangle-\sigma(s)}\right] = \mathbb{E}\left[e^{\langle s,\boldsymbol{y}\rangle}\right] e^{-\sigma(s)} = e^{\phi^*(\boldsymbol{\theta}+s)-\phi^*(\boldsymbol{\theta})-\sigma(s)} \\ &= \quad e^{\left[\phi^*(\boldsymbol{\theta}+s)-\sup_{\boldsymbol{y}\in\mathcal{B}}\langle s,\boldsymbol{y}\rangle\right]-\phi^*(\boldsymbol{\theta})} \end{aligned}$$

Now we tighten the exponent with respect to s as

$$\begin{aligned} \left[\phi^*(\boldsymbol{\theta}+s_*) - \sup_{\boldsymbol{y}\in\mathcal{B}} \langle s_*, \boldsymbol{y}\rangle\right] - \phi^*(\boldsymbol{\theta}) &= \inf_s \sup_{\boldsymbol{y}\in\mathcal{B}} [\phi^*(\boldsymbol{\theta}+s) - \langle s, \boldsymbol{y}\rangle] - \phi^*(\boldsymbol{\theta}) \\ &= \sup_{\boldsymbol{y}\in\mathcal{B}} \inf_s [\phi^*(\boldsymbol{\theta}+s) - \langle s, \boldsymbol{y}\rangle] - \phi^*(\boldsymbol{\theta}) \\ &= \sup_{\boldsymbol{y}\in\mathcal{B}} \langle \boldsymbol{y}, \boldsymbol{\theta}\rangle - \phi(\boldsymbol{y}) - \phi^*(\boldsymbol{\theta}) \\ &\geq \sup_{\boldsymbol{y}\in\mathcal{B}} \langle \boldsymbol{y}, \boldsymbol{\theta}\rangle - [\phi(\boldsymbol{y}') + \left\langle \boldsymbol{y}-\boldsymbol{y}', \nabla\phi(\boldsymbol{y}') + \delta(||\boldsymbol{y}-\boldsymbol{y}'||)\right\rangle] \\ &\quad - \phi^*(\boldsymbol{\theta}) \\ &= -\sup_{\boldsymbol{y}\in\mathcal{B}} \delta(||\boldsymbol{y} - (\nabla\phi)^{-1}(\boldsymbol{\theta})||) = -\sup_{\boldsymbol{y}\in\mathcal{B}} \delta(||\boldsymbol{y} - \mathbb{E}[Y]||) \end{aligned}$$

□

Appendix C

Proofs from Chapter 5

Notation: Follows notation of Chapter 5.

C.1 Proofs from Section 5.2.2

Proof of Lemma 18:

Proof. By Pinsker's inequality we have $\mathrm{KL}\left(\boldsymbol{p}\|\boldsymbol{q}\right) \geq 2\|\boldsymbol{p}-\boldsymbol{q}\|_1^2$.

$$\begin{aligned}\mathrm{KL}\left(\boldsymbol{p}\|\boldsymbol{q}\right) &= \sum_i p_i \log(\frac{p_i}{q_i}) \leq^{(a)} \sum_i p_i(\frac{p_i}{q_i}-1) \\ &= \sum_i \frac{p_i^2 - 2p_iq_i + q_i^2}{q_i} + \sum_i (p_i - q_i) \\ &\leq^{(b)} \frac{1}{\epsilon}\|\boldsymbol{p}-\boldsymbol{q}\|_2^2 \leq^{(c)} \frac{1}{\epsilon}\|\boldsymbol{p}-\boldsymbol{q}\|_1^2.\end{aligned}$$

Inequality (a) follows from $x - 1 > \log x$ and inequality b follows from $\min_i p_i \geq \epsilon$ and $\min_i q_i \geq \epsilon$. Combining upper and lower bounds we obtain $\frac{\mathrm{KL}(\boldsymbol{p}\|\boldsymbol{q})}{\mathrm{KL}(\boldsymbol{q}\|\boldsymbol{p})} < \frac{2}{\epsilon}$. □

Proof of Lemma 19:

Proof. It is required that $\hat{F}(\boldsymbol{\rho}^*, \boldsymbol{\rho}^*) \leq \hat{F}(\boldsymbol{\rho}^*, \tilde{\boldsymbol{\rho}}) + \frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{\rho}^*\|\tilde{\boldsymbol{\rho}}\right)$, using equation (5.8) we

obtain

$$\cancel{\hat{F}(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))} + \mathrm{KL}\left(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) \| \boldsymbol{\rho}^*\right) \leq \cancel{\hat{F}(\boldsymbol{\rho}^*, \boldsymbol{\rho}_*(\boldsymbol{\rho}^*))} + \mathrm{KL}\left(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) \| \tilde{\boldsymbol{\rho}}\right) + \frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{\rho}^* \| \tilde{\boldsymbol{\rho}}\right).$$

Re-arranging, we obtain that it is required that

$$\begin{aligned}
\frac{1-\beta}{\beta}\mathrm{KL}\left(\boldsymbol{\rho}^* \| \tilde{\boldsymbol{\rho}}\right) &\geq \mathrm{KL}\left(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) \| \boldsymbol{\rho}^*\right) - \mathrm{KL}\left(\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) \| \tilde{\boldsymbol{\rho}}\right) \\
&= \mathrm{KL}\left(\tilde{\boldsymbol{\rho}} \| \boldsymbol{\rho}^*\right) + \left\langle \overrightarrow{\rho_{*i} - \tilde{\rho}_i}, \overrightarrow{\log \frac{\rho_i^*}{\tilde{\rho}_i}} \right\rangle, \text{ or it is required that} \\
\frac{1-\beta}{\beta} &\geq^{(a)} \frac{\mathrm{KL}\left(\tilde{\boldsymbol{\rho}} \| \boldsymbol{\rho}^*\right)}{\mathrm{KL}\left(\boldsymbol{\rho}^* \| \tilde{\boldsymbol{\rho}}\right)} + \frac{\|\boldsymbol{\rho}_* - \tilde{\boldsymbol{\rho}}\|_2 \, \|\log(\boldsymbol{\rho}^*) - \log(\tilde{\boldsymbol{\rho}})\|_2}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|_2^2} \\
&\leq^{(b)} \frac{2}{\epsilon} + \frac{\delta}{\epsilon(1-\delta)}.
\end{aligned}$$

The first term in inequality (b) follows from lemma 18, the second term follows from the condition $\frac{\|\boldsymbol{\rho}_*(\boldsymbol{\rho}^*) - \tilde{\boldsymbol{\rho}}\|}{\|\boldsymbol{\rho}^* - \tilde{\boldsymbol{\rho}}\|} \geq \frac{\delta}{1-\delta}$ and the Lipschitz constant of $\frac{1}{\epsilon}$ of the vector valued function $\log(\cdot)$ on the set Δ_ϵ. To obtain inequality (a) we have used Cauchy-Schwarz, and lemma 18. □

C.2 Bregman-Affine Center

Since we will do a plugin replacement of KL divergence by a Bregman divergences in all of our cost functions, an optimization problem that will be of interest to us is that of minimizing over the second argument of a weighted sum of Bregman divergence from a set of points i.e.

$$\min_{\boldsymbol{y} \in \mathrm{int}(dom\phi)} \sum_i w_i D_\phi\big(\boldsymbol{x}_i \big\| \boldsymbol{y}\big) \quad s.t. \ \sum_i w_i \geq 0. \tag{C.1}$$

Our interest lies in the case where the summation of the weights are positive. The individual weights need not be positive. The minimizer of the problem will be termed the Bregman-Affine center of the vectors $\boldsymbol{x}_i$. To specify the solution of this problem we need to introduce

the notion of Legendre conjugates of convex functions.

Apart from playing a role in specifying the solution of the optimization problem (C.1) Legendre conjugates will find use in this paper to switch the order of the arguments in a Bregman divergence by drawing upon the identity

$$D_\phi\big(\boldsymbol{x}\Big|\Big|\boldsymbol{y}\big) = D_\psi\big(\nabla\phi(\boldsymbol{y})\Big|\Big|\nabla\phi(\boldsymbol{x})\big). \tag{C.2}$$

The RHS of (C.2) is of special consequence because minimizing it is equivalent to fitting $\{\nabla\phi(\boldsymbol{y})_i, \boldsymbol{x}_i\}_{1\leq i\leq n}$ by a Generalized Linear Model (GLM) with the canonical link function $\nabla\phi(\cdot)$. For the case of KL divergence the corresponding GLM is a logistic regression model.

With the necessary background in place, we state the following theorem regarding Bregman-Affine centers

Theorem 17. *Given a Bregman divergence $D_\phi\big(\cdot\Big|\Big|\cdot\big)$ defined by a convex function $\phi(\cdot)$ of Legendre type, $\boldsymbol{x}_i \in \operatorname{dom}\phi$ and $w_i \in \mathbb{R}$ s.t. the affine combination $\frac{\sum_i w_i\boldsymbol{x}_i}{\sum_i w_i} \in \operatorname{dom}(\phi)$, the problem*

$$\inf_{\boldsymbol{y}\in\operatorname{dom}\phi} \sum_i w_i D_\phi\big(\boldsymbol{x}_i\Big|\Big|\boldsymbol{y}\big) \quad \textit{s.t.} \ \sum_i w_i > 0 \tag{C.3}$$

has a minimizing sequence with a unique limit point $\boldsymbol{y}^* = \frac{\sum_i w_i\boldsymbol{x}_i}{\sum_i w_i}$, whereas the problem

$$\sup_{\boldsymbol{y}\in\operatorname{dom}\phi} \sum_i w_i D_\phi\big(\boldsymbol{x}_i\Big|\Big|\boldsymbol{y}\big) \quad \textit{s.t.} \ \sum_i w_i < 0 \tag{C.4}$$

has a maximizing sequence with a unique limit point $\frac{\sum_i w_i\boldsymbol{x}_i}{\sum_i w_i}$, and the set of limit point(s) $\boldsymbol{y}^*$ of the optimizing sequence of problem

$$\inf[\text{or},\sup]_{\boldsymbol{y}\in\operatorname{dom}\phi} \sum_i w_i D_\phi\big(\boldsymbol{x}_i\Big|\Big|\boldsymbol{y}\big) \quad \textit{s.t.} \ \sum_i w_i = 0 \tag{C.5}$$

satisfies

$$\nabla\phi(\boldsymbol{y}^*) = \operatorname{ArgSup}\left[\text{or, } \operatorname{ArgInf}\right]_{\boldsymbol{v}\in\operatorname{dom}(\phi*)} \left\langle \sum_i w_i \boldsymbol{x_i}, \boldsymbol{v} \right\rangle \tag{C.6}$$

Equation (C.6) *is a linear program with the optimum value*

$$-\,\delta^*{}_{\operatorname{dom}(\phi*)}\left([-]\sum_i w_i \boldsymbol{x_i}\right).$$

The solution set satisfies

$$\boldsymbol{y}^* \ni \begin{cases} \dfrac{\sum_i w_i \boldsymbol{x}_i}{\operatorname{Gauge}_{\operatorname{dom}(\phi)}\left(\sum_i w_i \boldsymbol{x}_i\right)} & \textit{if } \mathbf{0} \in \operatorname{dom}\phi \\ \\ \lim\limits_{c\to 0} \dfrac{\sum_i w_i \boldsymbol{x}_i}{c} & \textit{if the limit exists} \end{cases} \tag{C.7}$$

and lies on the boundary of $\operatorname{dom}(\phi)$.

Proof. Let $s = \sum_i w_i$, $\bar{\boldsymbol{x}} = \frac{\sum_i w_i \boldsymbol{x}_i}{\sum_i w_i}$ and $\bar{\phi} = \frac{\sum_i w_i \phi(\boldsymbol{x}_i)}{\sum_i w_i}$. We have

$$\begin{aligned} \sum_i w_i D_\phi\big(\boldsymbol{x}_i \big|\big| \boldsymbol{y}\big) &= \sum_i w_i D_\phi\big(\boldsymbol{x}_i \big|\big| \boldsymbol{y}\big) + s\phi(\bar{\boldsymbol{x}}) - s\phi(\bar{\boldsymbol{x}}) \\ &= s(\bar{\phi} - \phi(\bar{\boldsymbol{x}})) + s\phi(\bar{\boldsymbol{x}}) - s\phi(\boldsymbol{y}) - s(\bar{\boldsymbol{x}} - \boldsymbol{y})\nabla\phi(\boldsymbol{y}) \\ &= s(\bar{\phi} - \phi(\bar{\boldsymbol{x}})) + sD_\phi\big(\bar{\boldsymbol{x}} \big|\big| \boldsymbol{y}\big). \end{aligned} \tag{C.8}$$

The first term of RHS is a constant, and $D_\phi\big(\bar{\boldsymbol{x}} \big|\big| \boldsymbol{y}\big) \geq 0$ and $D_\phi\big(\bar{\boldsymbol{x}} \big|\big| \boldsymbol{y}\big) = 0 \iff \boldsymbol{y} = \bar{\boldsymbol{x}}$. If $\bar{\boldsymbol{x}}$ is on the boundary, consider any sequence $\lim_{t\to\infty} \boldsymbol{y}_t = \bar{\boldsymbol{x}}$. Using property **P2** we obtain $\lim_{t\to\infty} D_\phi\big(\bar{\boldsymbol{x}} \big|\big| \boldsymbol{y}_t\big) = 0$, hence $\boldsymbol{y}_t$ is a minimizing sequence. This proves (C.3) and (C.4). The special case of this theorem for $\sum_i w_i = 1$ was proven by Banerjee et al. (2005) as well as the proposition that Bregman divergences are the only cost function for which the property is true.

For the remaining, consider $s = 0$. In this particular case equation (C.8) is no longer

valid because it requires $\bar{\boldsymbol{x}}$ to be well defined (whereas it is not because of division by zero). However, we have the following relation:

$$\sum_i w_i D_\phi\big(\boldsymbol{x}_i \big\| \boldsymbol{y}\big) = s\bar{\phi} - \left\langle \left(\sum_i w_i \boldsymbol{x}_i \right), \nabla\phi(y) \right\rangle .$$

Expression (C.6) follows from the observation that the first term is constant and that $\nabla\phi(y) \in \mathrm{dom}(\phi^*)$ by definition. This domain transformation is critical in converting a non-linear problem into the linear programming problem (C.6).

In what follows we elaborate on the minimization part of the problem (C.5) because it applies directly to our consensus ranking problem, the maximization can be handled similarly.

$$\inf_{\boldsymbol{v}\in\mathrm{dom}(\phi*)} -\left\langle \sum_i w_i \boldsymbol{x_i}, \boldsymbol{v} \right\rangle = -\sup_{\boldsymbol{v}\in\mathrm{dom}(\phi*)} \left\langle \sum_i w_i \boldsymbol{x_i}, \boldsymbol{v} \right\rangle \triangleq -\,\delta^*{}_{\mathrm{dom}(\phi*)} \left(\sum_i w_i \boldsymbol{x_i} \right). \tag{C.9}$$

The solution of (C.9) is the point or a face of $\mathrm{dom}\,\phi^*(\cdot)$ exposed by the direction $\sum_i w_i \boldsymbol{x_i}$. To obtain a solution (C.7) we use a sequence of unconstrained optimization problems.

The constraint $\boldsymbol{v} \in \mathrm{dom}(\phi*)$ is replaced by an appropriate barrier function $B(\boldsymbol{v})$ that enforces the constraint. By definition the barrier function has to satisfy

$$\lim_{v\to\mathrm{bd}(\mathrm{dom}(\phi*))} B(\boldsymbol{v}) = \infty \quad \text{and} \lim_{v\to\mathrm{bd}(\mathrm{dom}(\phi*))} \nabla B(\boldsymbol{v}) = \infty.$$

Both these properties are satisfied by the function $\phi^*(\cdot)$, because $\phi(\cdot)$ and consequently (Rockafellar, 1996) $\phi^*(\cdot)$ is a Legendre function. This allows us to use it as a barrier function that is naturally suited to the problem. As a result, we obtain the modified sequence of optimization problems defined for each value of c_t that satisfies the condition $\lim\, c_t \downarrow 0$:

[1]

$$\max_{\boldsymbol{v}} \left\langle \sum_i w_i \boldsymbol{x_i}, \boldsymbol{v} \right\rangle - c_t \phi^*(\boldsymbol{v}) \triangleq c_t \phi \left(\frac{\sum_i w_i \boldsymbol{x_i}}{c_t} \right). \tag{C.10}$$

A point in the solution set can be computed as the limit of the solutions of the sequence of Legendre dual evaluations (C.10), and is given by

$$\boldsymbol{v}^*(c_t) = \nabla^{-1} \phi^* \left(\frac{\sum_i w_i \boldsymbol{x_i}}{c_t} \right) = \nabla \phi \left(\frac{\sum_i w_i \boldsymbol{x_i}}{c_t} \right).$$

Following which, we obtain

$$\boldsymbol{y}^* \ni \lim_{c_t \to 0} (\nabla)^{-1} \phi(\boldsymbol{v}^*(c_t)) = \lim_{c_t \to 0} (\nabla)^{-1} \phi \left(\nabla \phi \left(\frac{\sum_i w_i \boldsymbol{x_i}}{c_t} \right) \right).$$

The transformed optimization problem is solved for a reducing sequence of c_t such that the solution $\boldsymbol{v}^*$ lies in the closure $\mathrm{cl}\,\mathrm{dom}\,\phi^*$. Thus, from the relation

$$\lim_{t \to \infty} c_t = \sup \left\{ c \,\middle|\, \frac{\sum_i w_i \boldsymbol{x_i}}{c} \in \mathrm{dom}\,\phi \right\}$$

we obtain from the definition of gauge that $\lim_{t \to \infty} c_t = \mathrm{Gauge}_{\mathrm{dom}(\phi)} \left(\sum_i w_i \boldsymbol{x}_i \right)$. □

Theorem (17) plays a critical role in the rest of the paper, therefore we briefly summarize its significance which spans both the theoretical and the computational. The parts (C.3) and (C.4) have several important consequences. The first is that the nonlinear non-convex cost function has not only a unique solution but also that can be computed in a simple closed form. Furthermore the solution has the simple form of an affine combination of the vectors $\boldsymbol{x}_i$ combined according to the normalized weights $\frac{\boldsymbol{w}}{\sum_i w_i}$.

Even more strikingly, Bregman divergences are the only divergences for which such

[1] One would recognize that the extreme RHS of equation (C.10) is the limiting case of the dilation function of $\phi(\cdot)$. The interplay between the support function and the barrier function should not be surprising because the Legendre dual of the support function is the indicator function, which in this case is approximated by the barrier function. Positive multiples of the barrier function serves as a differentiable and a convergent approximation to the indicator function $\delta(\cdot | \,\mathrm{dom}\,\phi^*)$. The optimal $\boldsymbol{y}$ is obtained by inverting the domain transform $\nabla\phi(\boldsymbol{y}^*) = \boldsymbol{v}^*$ to obtain the relation (C.7).

an affine combination is the solution. The special case where $\sum_i w_i = 1$ has been shown by Banerjee et al. (2005), in this case the affine combination reduced to a simple convex combination. Since the affine combination subsumes convex combination, it follows directly that Bregman divergences are the only class for which the optimum is obtained at the affine center. The results (C.3) and (C.4) extend the results obtained by Banerjee et al. (2005) to the cases $\sum_i w_i > 0$ and $\sum_i w_i < 0$. We however lose some universality compared to the convex case because the previous result (Banerjee et al., 2005) holds for any set of vectors $\boldsymbol{x}_i$ in the domain of the Bregman divergence whereas when $\sum_i w_i$ is higher or lower than 1, the result applies to the subset such that the affine combination of $\boldsymbol{x}_i$ by the weights $\frac{\boldsymbol{w}}{\sum_i w_i}$ lie in the domain of the Bregman divergence.

The extra requirements on $\boldsymbol{x}_i$ has important practical consequences because it might be difficult to guarantee that the vectors $\boldsymbol{x}_i$ satisfy the condition required, especially if the vectors $\boldsymbol{x}_i$ are an intermediate quantity in a series of computations. However, if the relative interior of the domain of $\phi(\cdot)$ spans its entire affine hull, no such extra conditions need to be checked.

For the purpose of this paper, the role played by part (C.5) of theorem (17) is crucial. Although the closed form solutions of the problems (C.3) and (C.4) become degenerate at $\sum_i w_i = 0$, part (C.5) shows that the optimization problem may still be well defined. It turns out that the solution in this case can not only be defined but unlike parts (C.3) and (C.4), it requires no extra conditions on $\boldsymbol{x}_i$.

As a consequence of (C.5), first we are able to reduce the non-linear problem to an equivalent linear program by domain transformation. This is no doubt an important simplification but unless carried through further it would have entailed steep computational expenses. For example, if any algorithm requires a solution of the optimization problem (C.5) in a repeated intermediate step, that would have required numerically solving several inner linear programming problems. The striking feature of (C.5) is that the resulting linear programming problem affords a closed form solution.

Bibliography

http://webspam.lip6.fr. 2007.

S. Acharyya, O. Koyejo, and J. Ghosh. Learning to rank with Bregman divergences and monotone retargeting. In *Uncertainty in Artificial Intelligence, UAI 2012*, 2012.

Sreangsu Acharyya and Joydeep Ghosh. Outlink estimation for pagerank computation under missing data. In *Thirteenth International World Wide Web Conference WWW 2004*, 2004.

Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In *Conference on Learning Theory, COLT 2008*, pages 87–98, 2008.

Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *NIPS*, pages 316–322, 1995.

Ricardo Baeza-Yates, Paolo Boldi, and Carlos Castillo. Generalizing pagerank: Damping functions for link-based ranking algorithms. In *Proceedings of ACM SIGIR*, pages 308–315, Seattle, Washington, USA, August 2006. ACM Press. URL `http://dx.doi.org/10.1145/1148170.1148225`.

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 020139829X.

Gükhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007. ISBN 0262026171.

A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

R. E. Barlow and H. D. Brunk. The isotonic regression problem and its dual. *Journal of American Statistical Association*, 67(337):140–147, 1972.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439, 1990.

R. Blahut. Computation of channel capacity and rate-distortion functions. *Information Theory, IEEE Transactions on*, 18(4):460–473, 1972.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

Sergei Brin and Larry Page. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. URL `http://citeseer.ist.psu.edu/page98pagerank.html`.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 129–136, New York, NY, USA, 2007a. ACM.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, New York, NY, USA, 2007b. ACM.

Y. Censor and T. Elfving. A multiprojection algorithm using Bregman projections in a product space. *Numerical Algorithms*, 8:221–239, 1994.

Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34(3):321–353, 1981.

Y. Censor and S. Zenios. The proximal minimization algorithmwith D-functions. *Journal of Optimazation Theory and Applications*, 73:451–464, 1992.

Yair Censor. Row-action methods for huge and sparse systems and their applications. *SIAM Review*, 23:444–466, 1981.

Yair Censor and Stavros Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997. ISBN 019510062X.

Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

William. Cohen, Robert Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

I. Csiszar. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

Frank Deutsch and Hein Hundal. The rate of convergence for the cyclic projections algorithm I. *Journal of Approximation Theory*, 142:36–55, 2006.

O. Devolder, F. Glineur, and Yu. Nesterov. First order methods of smooth convex optimization with inexact oracle. In *CORE Discussion Paper 2011/2*, 2011.

Persi Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

J. Douceur. The sybil attack. In *Proceedings of First International Peer to Peer Systems Workshop*, 2002.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0. doi: http://doi.acm.org/10.1145/371920.372165.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003. ISSN 1533-7928.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *COLT*, pages 29–40, 1999.

S.J. Grotzinger and C. Witzgall. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12:247–270, 1984.

Peter Grunwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *The Annals of Statistics*, 32:1367–1433, 2004.

Peter Grunwald and A. P. Dawid. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61:151–165, 2005.

Zoltan Gyongyi, Garcia-Molina Hector, and Jan Pedersen. Combating web spam with trustrank. In *VLDB*, pages 576–587, 2004.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2007.

David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10:1705–1736, 2009.

Monika R. Henzinger. Algorithmic challenges in web search engines. *Internet Math.*, 1(1): 115–123, 2003.

William Hersh, Chris Buckley, T. J. Leone, and David Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 192–201, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

Alfredo N Iusem. Steepest descent methods with generalized distances for constrained optimization. *Acta Applicande Mathematicae*, 46:225–246, 1997.

Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM. ISBN 1-58113-226-3. doi: 10.1145/345508.345545. URL `http://doi.acm.org/10.1145/345508.345545`.

Sham Kakade, Adam Kalai, Varun Kanade, and Ohad Shamir. Efficient learning of generalized and single index models with isotonic regression. In *ARXIV:1104.2018v1 [cs.AI]*, 2011.

Adam Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009.

Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon. Fast projection-based methods for the least squares nonnegative matrix approximation problem. *Stat. Anal. Data Min.*, 1(1): 38–51, 2008.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132, 1995.

K. C. Kiwiel. Generalized Bregman projections in convex feasibility problems. *Journal of Optimization Theory and Applications*, 96:139–157, 1988.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999a. URL `citeseer.nj.nec.com/kleinberg97authoritative.html`.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999b.

A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised rank aggregation with domain-specific expertise. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 7 2009.

Vijay Krishnan and Rashmi Raj. Web spam detection with anti-trust rank. In *AIRWeb*, pages 37–40, 2006.

Yanyan Lan, Tie-Yan Liu, Zhiming Ma, and Hang Li. Generalization analysis of listwise learning-to-rank algorithms. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 577–584, New York, NY, USA, 2009. ACM.

Guy Lebanon and John D. Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *ICML*, pages 363–370, 2002.

E. H. Lehmann. *Theory of Point Estimation*. John Wiley & Sons, 1983.

Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *In Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

Zhi-Quan Luo. On the linear convergence of the alternatind direction method of multipliers. In *ARXIV:1208.3922v1[math.OC]*, 2012.

A. Mccallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 662–667, 1999.

C. E. McCulloch and S. R. Searle. *Generalized Linear and Mixed Models*. John Wiley & Sons, 2001.

Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *ICML*, 2012.

Marc A. Najork, Hugo Zaragoza, and Michael J. Taylor. Hits on the web: how does it compare? In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 471–478, 2007.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A, General*, 135:370–384, 1972.

Arkadi Nemirovski. *Lectures on modern convex optimization*. Society for Industrial and Applied Mathematics, 2001.

Yu. Nesterov. Universal gradient methods for convex optimization problems. In *CORE Discussion Paper 2013/26*, 2013.

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, March 2008. ISSN 1542-7730. doi: 10.1145/1365490.1365500. URL `http://doi.acm.org/10.1145/1365490.1365500`.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, 1997. ISSN 0162-8828.

Tao Qin, Xiubo Geng, and Tie-Yan Liu. A new probabilistic model for rank aggregation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1948–1956, 2010.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

Pradeep Ravikumar, Ambuj Tewari, and Eunho Yang. On NDCG consistency of listwise ranking methods. In *Proceedings of 14th International Conference on Artificial Intelligence and Statistics*, AISTATS, 2011.

M. Reid and R. Williamson. Surrogate regret bounds for proper losses. In *ICML*, 2009.

Robert.M.Solow and Paul.A.Samuelson. Balanced growth under constant returns to scale. *Econometrica*, 21(3):412–424, 1953.

R T. Rockafellar. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, December 1996. ISBN 0691015864.

Walter Rudin. *Principles of Mathematical Analysis*, pages 203,318. McGraw-Hill Book Company, 1976.

Shai Shalev-Shwartz and Sham M. Kakade. Mind the duality gap: Logarithmic regret algorithms for online optimization. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 1457–1464. MIT Press, 2008.

Panayiotis Tsaparas. Link analysis ranking. In *http://www.webir.org/resources/phd/Tsaparas_2004.zip*, page Chapter 3, 2004.

P. Tseng. Convergence of block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:473–492, 2001.

Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998. ISBN 978-0-471-03003-4.

Jason Weston and John Blitzer. Latent structured ranking. In *Uncertainty in Artificial Intelligence, UAI 2012*, 2012.

B. Wu, V. Goel, and B.D. Davison. Propagating trust and distrust to demote web spam. In *Proc. Models of Trust for the Web Workshop (MTW), International Wor ld Wide Web Conference*, 2006.

L. Yeganova and W.J. Wilbur. Isotonic regression under Lipschitz constraints. *Journal of Optimization Theory and Applications*, 141:429–443, 2009.

W. I. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice Hall, 1969.